MASTER OF SCIENCE IN
DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK
DISSERTATION

UNIVARIATE TIME SERIES FORECASTING: COMPARING ARIMA & LSTM NEURAL NETWORK TO THE RANDOM WALK BENCHMARK FOR EXCHANGE RATES

RICARDO SÁNCHEZ GAVILANES

MARCH - 2022

MASTER OF SCIENCE IN
DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK
DISSERTATION

UNIVARIATE TIME SERIES FORECASTING: COMPARING ARIMA & LSTM NEURAL NETWORK TO THE RANDOM WALK BENCHMARK FOR EXCHANGE RATES

RICARDO SÁNCHEZ GAVILANES

SUPERVISOR:
NUNO SOBREIRA

MARCH - 2022

Abstract

The difficulty of forecasting Exchange Rates has been a longstanding problem for economists and data analysts around the world. Nevertheless, a model that could produce accurate forecasts and outperform the random walk (RW) benchmark would be beneficial to policymakers and investors as it might help mitigate the effects of inflation, thus, having a real impact on the economic perspective.

The objective of this paper is to develop and analyze the results of an ARIMA and LSTM models and determine if univariate time series models can show an improved accuracy at a 5-day and 60-day time horizon compared to the driftless random walk (DRW) model, which is the proposed benchmark in this study. In order to perform this analysis, daily exchange rate data for the currency pair USD/EUR was retrieved from the United States Board of Governors of the Federal Reserve System download data program, and later cleaned and manipulated using Python to produce forecasts for each of the models.

The predictive accuracy was calculated, and the errors were measured with the MAPE, RMSE, and MSE metrics. Among the three models tested, I concluded that both the LSTM and ARIMA models outperformed the DRW benchmark at the 60-day horizon, which adds evidence of the suitability of autoregressive and machine learning univariate models when forecasting exchange rates, and of their potentially superior performance compared to classical models based on economic fundamentals, which have traditionally failed to outperform the RW benchmark. On the contrary, in the 5-day horizon, the DRW model showed the highest predictive accuracy, thus supporting the existing literature in the difficulty of forecasting exchange rates at short horizons.

Keywords: Exchange rate forecasting, univariate time series, ARIMA, LSTM, RNN, Driftless Random Walk.

Acknowledgements

This dissertation is the final result of a period filled with hard work, discipline, and constant learning which I couldn’t have gone through without the support of the most important people in my life, which I would like to thank immensely. To my partner Fina, who inspires me every day to be a better person and fills my life with joy. To my brother and sister, Luis and Paola, who have been by my side in the best and worst moments, and have supported me in every decision I’ve made, thank you for being my best friends. To my father, Felipe, whose patience, wisdom, and support have guided me through very confusing times and who will always be my role model. And to my mother, Fanny, who is the inspiration for every achievement in my life and whose love is always and will always be with me wherever I go.

Additionally, I would like to thank to all my professors in this Masters’ program, which have taught me a lot and helped me give my career a new direction, and specially to my supervisor, Nuno Sobreira, whose comments, and suggestions were really helpful when writing this dissertation.

Table of Contents

List of Tables

List of Figures

1. Introduction

Exchange Rate is defined as the rate at which one currency will be exchanged by another. In effect, it compares the value of a country's currency compared to another country or economic zone. It is a key component of a country's trade policy; whether determined by extrinsic events or internal policies, a high volatility in the valuation of a currency will most certainly impact international trade, which concurrently plays a vital role in overall economic performance. Therefore, monetary policymakers are aware that if considerable nominal exchange rate fluctuations persist over time, real exchange rate misalignments may develop which could have significant implications for a country's economic outlook. Due to this, central banks must carefully monitor the evolution of exchange rates as, in fact, the market value of a currency is fundamental for understanding the medium and long-term inflation outlook through its impact on the economy.

For the aforementioned reasons, it proves very useful to be able to anticipate future exchange rate fluctuations. However, the difficulty of forecasting exchange rates has been a longstanding problem in international economics and exchange rate forecasting models have generally exhibited poor out-of-sample performances. Moreover, even though models have shown some forecasting power at horizons of two to four years (Meese and Rogoff, 1983b; Mark, 1995; Engel, and West, 2007), attempts to predict exchange rates at horizons, which are more relevant to policymakers, of one month to one year, have been far less successful.

Overall, we can differentiate the existing empirical approaches into two categories: First, economic-based models which try to exploit the influence of macroeconomic variables on the exchange rate markets, and second, non-theoretical models which do not take economic factors into account. Several exchange rate forecasting models have been derived from the former. However, while many of these models might be theoretically sustained, they have exhibited low out-of-sample performance and have been unable to deliver more accurate nominal exchange rate forecasts than the naïve prediction associated with the random walk (RW) model, suggesting that exchange rates are largely disconnected from economic fundamentals at short horizons (Meese and Rogoff 1983b). This conclusion is better known in international economics as "the Meese and Rogoff puzzle".

Some of the leading exchange rate forecasting models, which are based on economic theories, are the Purchase Power Parity (PPP) model, the Monetary Model (MM) and the Uncovered Interest Parity (UIP) model. Historically, these models failed to demonstrate significant influence of the economic fundamentals used to predict exchange rate movements in the short term, but this does not necessarily imply the inexistence of a relationship between them. One critical factor in the functional form of these models, as shown by Amat et al. (2018), is the strict linearity of the equations; therefore, potential nonlinearities have been deliberately not considered. This means that by applying nonlinear models, economic fundamentals could show significant effects on exchange rates and the performance of the models could notably improve (Haggan & Ozaki, 1981).

Machine learning algorithms are useful for implementing non-linear models and in recent years have been gaining popularity due to large improvements in computing power even though their development is not recent. These models come with a high number of parameters which usually leads to a trade-off between improved accuracy and low interpretability.

The use of machine learning methods have already been tested in the context of currency exchange prediction; most notably, the aforementioned paper by Amat et al. (2018) used simple machine learning (ML) models with economic fundamentals to test its results against ordinary least squares (OLS) estimates. Even though their ridge regression and exponentially weighted average models improve the performance compared to the OLS estimates, they occasionally failed to improve the accuracy of the RW model, therefore adding evidence against the relationship of economic factors with the variability of exchange rates. Consequently, several researchers have developed forecasting algorithms that do not consider economic variables, but rather past observations only as a univariate time series.

There are some studies that have used the ARIMA and LSTM class of models for FX forecasting; for instance, Zhang (2007) compared the performance of ARIMA with simple linear models, whereas Rout et al. (2014) used an Autoregressive Moving Average (ARMA) model with Differential Evolution (DE) techniques to forecast 3 currency pairs. Additionally, Qu & Zhao (2019) compared the accuracy of the LSTM neural network against a vanilla recurrent neural network (RNN), to name a few examples.

With the abovementioned context, the objective of this paper is to compare the predicting accuracy of an autoregressive integrated moving average (ARIMA) and long short-term memory neural network (LSTM) with the random walk (RW) benchmark in the context of univariate time series, and by doing so, contribute to the existing literature and increase the interest of applying data analytics techniques and machine learning algorithms to forecast exchange rates. The study made by Rossi (2013) implies that the driftless random walk (DRW) is the toughest model to beat when forecasting exchange rates, which is the main reason it's used as a benchmark in this investigation.

This paper is structured in the following way: in the first section the literature of exchange rate forecasting using economic fundamentals is reviewed, followed by investigations focused on the development and application of ARIMA and Machine Learning algorithms to exchange rate time series. In the second section, the methodological framework is introduced which include the theoretical foundations of the ARIMA and LSTM models, the random walk benchmark and the techniques used for the models' performance evaluation and the metrics applied to calculate their errors. The third section includes a description of the dataset and the tools used for the development of the models and the analysis of the data, and in the fourth section the results are presented. Finally, the conclusions of this comparative analysis are drawn in the fifth section.

2. Literature review

2.1. Models based on economic fundamentals

Exchange rate forecasting models based on economic fundamentals attempted to establish a connection between the evolution of exchange rates and the variability of fundamentals such as the theory of Purchasing Power Parity (PPP), the Uncovered Interest Rate (UIP) and the Monetary Model (MM). These models became increasingly popular in the second half of the 20th century, following the breakdown of the Bretton Woods fixed exchange rate system.

The first FX analytical model was proposed by Mundell (1968) and Fleming (1962), which was later extended by Dornbusch (1976) to the sticky prices model. Later, the flexible price monetary model developed by Bilson (1978), Stockman (1980) and Lucas (1982), which was extended by Frankel (1979) to its real interest rate differential variant and has been regarded for many years as the central work of exchange rate economics. These models are mainly used for medium and long-term forecasting.

Further developments of empirical models to forecast exchange rates were done in studies made by Cornell (1977), Mussa (1979) and Frenkel (1981), they all agree in two main points:

1. Any correlation found in exchange rates by in-sample tests will most probably be unstable over time.
2. The natural logarithm of the spot exchange rate follows approximately a random walk (RW) model.

The results of the tests performed in these studies led to the conclusion that exchange rates are largely unpredictable, but it paved the way for studies that would utilize the RW model as a benchmark for model accuracy.

Additionally, Meese and Rogoff (1983b) in their seminal paper, compare out-of-sample accuracy of various structural and time series exchange rate models. They included both sticky-price as well as flexible-price models following the Drenkel–Bilson, Dornbusch–Frenkel and Hopper–Morton monetary approaches. By generating forecasts at one-to-twelve-month horizons for different currency pairs and trade-weighted dollar exchange rates, the authors showed that none of the used models generated any improvement in root mean square error (RMSE) or mean absolute error (MAE) over the random walk model, even when having the advantage of using known economic fundamentals.

The paper by Mark (1995), showed that a model based on MM fundamentals generally outperformed the Driftless Random Walk (DRW) benchmark in forecasting exchange rate differentials at larger horizons. On the contrary, the findings by Engel et al. (2017) exhibited irregular accuracy and could not consistently beat the DRW benchmark in out-of-sample forecasts at short horizons, despite the fact that the coefficients of fundamentals in their OLS regression were statistically significant.

Additional papers have reproduced similar results. Vitek (2005) also considered both sticky-price and flexible-price versions of the monetary model of nominal exchange rate determination using a larger dataset and, in agreement with the existing empirical literature, they concluded that nominal exchange rate fluctuations are, to a large extent, difficult to forecast and that the DRW generally outperforms the monetary model in terms of predictive accuracy at all horizons.

On the other hand, there have been several studies which attempted to exploit nonlinearities in exchange rates to forecast future log exchange rates with rather limited success. For example, Engel & Hamilton (1990) analyzed four years' quarterly out-of-sample point predictions of their segmented-trends model for the pound, the Deutsche mark, and the French franc. The results of their model are outperformed by the RW in the four-quarter horizon.

Another example is the exponential autoregressive (EAR) model originally proposed by Haggan & Ozaki (1981) and later revisited by Granger & Terasvirta (1993) and Terasvirta (1994) in which deviations from PPP are shown to follow a non-linear process that is mean reverting with the speed of adjustment towards equilibrium varying directly with the extent of the deviation from PPP.

Michel et al. (1997) proposed a framework in for the empirical analysis of PPP that allows for the fact that commodity trade is not frictionless, meaning that as a result of the costs of trading goods, persistent deviations from PPP are implied as an equilibrium feature of the exchange rates models. In their study, they were able to reject linearity in favor of an EAR process.

More recently, the influential article by Rossi (2013) reiterated previous results that confirms the difficulty of predicting exchange rates with systematically better results than by using the DRW model at short horizons. Cheung et al. (2019) answer the question "Are exchange rates predictable?" by suggesting that it largely depends on the choice of predictor, forecast horizon, sample period, model, and forecast evaluation method. They conclude that predictability is most apparent when one or more of the following holds:

- the predictors are Taylor rule and net foreign assets fundamentals,
- the model is linear, and

- a small number of parameters are estimated.

By incorporating economic fundamentals such as PPP, UIP, and MM models, Amat et al. (2018) used the Exponentially Weighted Average (EWA) strategy with discount factors and ridge regression to forecast exchange rates. The authors compare the out-of-sample predictions of their models with OLS and DRW benchmark models with limited success; some tests beat the DRW model by less than one percent of the root mean square error for different country pairs. Their directional forecasts show that the EWA model performs best against the OLS regression, but the latter can also correctly predict the direction of exchange rate movement more than half of the time for most currency pairs.

2.2. The Driftless Random Walk Benchmark

As it is widely recognized by economic scholars, the simple Driftless Random Walk (DRW) is a very difficult benchmark to beat in out-of-sample comparisons for exchange rates, as shown initially by Meese and Rogoff (1983a, 1983b). Several studies have attempted to challenge the results of this seminal research, with varying success, more specifically:

- FX differences are leptokurtic (Hsieh, 1988)
- ANNs, including hybrid ANNs can yield a higher profit than linear models (Refenes, 1992; Lee and Jhee, 1994; Zhang, 1994)
- Technical analysis is widely used by many investors (Hawley et al., 1993; Mehta, 1995)

On top of that, several papers have claimed that even the narrow existing evidence that support forecasts of the exchange rates based on economic fundamentals is either very sensitive to the choice of the sample and the data antiquity, not robust to the inference procedures used, or rather weak (Kilian, 1999; Berben & Van Dijk, 1998; Groen, 1999; Berkowitz & Giorgianni, 2001; Faust, Rogers & Wright, 2003).

On the other hand, many researchers have tried different approaches to tackle exchange rate forecasting, using the RW model as benchmark. Carriero, Kapetanios & Marcellino (2008) forecasted exchange rates with a large Bayesian VAR (BVAR) model, using a panel of 33 exchange rates vis-à-vis the US Dollar. After producing forecasts for all the exchange rates, their model performed consistently better than the Driftless Random Walk (DRW) for most of the countries, at any forecast horizon.

Pincheira & Neumann (2018) examined the accuracy of survey-based expectations of the Chilean exchange rate relative to the US dollar. Their results showed that survey-based forecasts can outperform the Driftless Random Walk (DRW) in terms of Mean Squared Prediction Error at several forecasting horizons.

2.3. ARIMA-based models

There is extensive literature of exchange rate forecasting using ARIMA as the model of choice. Among others, Weisang & Awazu (2014) developed and tested the Autoregressive Integrated Moving Average (ARIMA) model to forecast exchange rates using daily and monthly information as the variable output. This study suggested that the ARIMA model is a comparatively accurate model to forecast exchange rates.

Kamruzzaman & Sarker (2004) used three neural-based forecasting algorithms to predict six currency pairs. Using moving average technical indicators, they compared the results with those of the traditional ARIMA model; they showed that the neural-based models produced forecasts closer to the actuals.

Babu and Reddy (2015) examined the performance of ARIMA, Neural Network and Fuzzy neuron models in forecasting the currencies traded in Indian foreign exchange markets. Daily RBI reference exchange rates from January 2010-April 2015 were used for the analysis. The results of this investigation conflicted with earlier studies (Chattopadhyay & Chattopadhyay S 2012) which concluded that neural network models perform better than ARIMA model and fuzzy neuron model performs better than neural network model.

Durat & Echeverria's (2002) study showed the ARIMA model performs better than complex nonlinear models when predicting exchange rates. Ayodele et al. (2012) compared Neural Networks (NN) models with ARIMA to predict stock prices and revealed that NN models outperform the ARIMA model.

Kaynar and Taştan (2009), used both daily and monthly rates from January 2000 to June 2008 to find the best model to predict the currency pair USD/TRY. They showed that the best model for daily exchange rates is ARIMA (2,1,0) and for monthly exchange rates ARIMA (0,1,1).

Özkan (2011), using 338 monthly observations for the years of 1986 and 2010 for the currency pair USD/TRY, compared three models including ARIMA, ANN and Conjunctural Model (CM). She concluded that ANN is the most successful method for predicting USD/TRY exchange rate.

2.4. Machine Learning-based models

The inclusion of machine learning algorithms in classical economic models has been proved beneficial to the overall out-of-sample accuracy. Bajari et al. (2015) compared random forests and support vector machines with classical econometric models for demand forecasting. The authors demonstrated that model combination with linear regression improves out-of-sample forecasting accuracy; their machine learning models delivered consistently better predictions that purely economic-based models.

Plakandaras, Papadimitriu, and Gogas (2017) combined signal processing to machine learning methodologies by introducing a hybrid Ensemble Empirical Mode Decomposition (EEMD), Multivariate Adaptive Regression Splines (MARS) and Support Vector Regression (SVR) model in order to forecast the monthly and daily exchange rates for five currency pairs. This implementation outperformed the forecasting ability in exchange rate forecasting and high Sharpe Ratios compared to various models taking data snooping bias in consideration, rejecting the Efficient Market Hypothesis for all foreign exchange markets.

Amat, Michalski, and Stoltz (2018) applied sequential ridge regression and the exponentially weighted average strategy which resulted in improved exchange rate forecasts for major currencies over the period 1973–2014 at a 1-month forecast horizon using fundamentals from simple exchange rate models (PPP or UIRP) or Taylor-rule based, which beat the no-change forecast.

Qu and Zhao (2019) compared the evaluation indexes of two deep learning models. The empirical results showed that the LSTM neural network model has smaller root mean square error (RMSE) and mean absolute error (MAE) than the RNN network model, therefore, higher accuracy. In addition, Yldırım, Toroslu, and Fiore (2021), utilized macroeconomic data and technical indicator data to propose a hybrid model, which combined two separate LSTMs corresponding to these two data sets. The hybrid was found to be quite successful in out-of-sample experiments.

Abedin, Moon, Hassan, & Hajek (2021) predicted the exchange rates of 21 currencies against the USD during the pre-COVID-19 and COVID-19 periods by integrating a Bagging Ridge (BR) regression with Bi-directional Long Short-Term Memory (Bi-LSTM) neural networks as base regressors. To demonstrate the effectiveness of this Bi-LSTM BR approach, they compared the prediction performance with several other traditional machine learning algorithms, such as the regression tree, support vector regression, and random forest regression, and deep learning-based algorithms such as LSTM and Bi-LSTM. Their integrated approach outperformed the compared models in forecasting exchange rates in terms of prediction error. Nevertheless, the performance of their model showed high variance during non-COVID-19 and COVID-19 periods across currencies.

Moreover, the recent study by Pfahler (2021) used monthly data from 1973 to 2014 for ten currency pairs of OECD countries to produce out-of-sample forecasts with artificial neural networks (ANN) and XGBoost models and compared it to the RW benchmark. For the prediction of exchange rate differentials, no convincing evidence regarding the predictive power of economic fundamentals was found, reproducing the results reached by the Meese and Rogoff puzzle. Nevertheless, the prediction of exchange rate movement directions, the XGBoost models were able to outperform the RW benchmark by a small margin, which was in some cases statistically significant. The ANN models were able to outperform the random walk benchmark by a larger margin when time dummies were included in the models which led him to conclude that complex interactions between time dummies and fundamentals could be the key to good directional forecasts for exchange rates.

3. Methodology

Due to its efficiency and adaptability, Python was chosen as the preferred programming language to process the data and to develop all the models used in this paper. Moreover, the plotly library was used to produce the graphics, and the metrics from the sklearn library to calculate the accuracy of the models. Finally, the tensorflow keras and the statsmodels libraries were used to produce the LSTM and ARIMA models respectively.

3.1 ARIMA

The ARIMA class provides a very general class of univariate models that can produce competitive predictions for the time series of interest. It characterizes and models temporal dependence between observations via autocorrelations in the data and uses that information to fit a final model. The methodology of ARIMA starts by making the series stationary by differencing, and in addition, differencing can eliminate possible trends and seasonality such that the mean is stabilized. Other transformations such as logarithms can help stabilize the variance of a time series. Lags of the series (after making it stationary) in the ARIMA equation are called "autoregressive" (AR) terms, lags of the 1- step ahead forecast errors are called "moving average" (MA) terms, and a time series which needs to be differenced to be made stationary is said to be an "integrated" (I) time series. The time series behavior is described by the following equation:

$$\Delta^d Y_t = \phi_0 + \phi_1 \Delta^d Y_{t-1} + \cdots + \phi_p \Delta^d Y_{t-p} + \mathcal{E}_t + \theta_1 \mathcal{E}_{t-1} + \cdots + \theta_q \mathcal{E}_{t-q},$$

where d is the number of times we need to take first differences to the variable Y_t to make the series stationary.

A weakly stationary time series is one which mean, variance, and autocovariances do not depend on the time at which the series is observed. For example, a white noise series is stationary; it doesn't matter at which point in time we observe it, it remains the same. On the contrary, series with seasonal or trend components are not stationary since trend and season will at least affect the mean values at different times. Formally, a time series process $\{Y_t\}$ is weakly stationary if:

$$E(Y_t) = \mu < \infty,$$

$$Var(Y_t) = E[(Y_t - \mu)^2] = \gamma_0 < \infty,$$

$$Cov(Y_t, Y_{t-k}) = E[(Y_t - \mu)(Y_{t-k} - \mu)] = \gamma_k,$$

which means that the mean and variance of Y_t need to be constant over time and the covariance between (Y_t, Y_{t-k}) must also be constant and, at most, may be a function of k. Afterwards, the conditional mean model for the given data is identified (Olajide et al. 2012).

ADF (Augmented Dickey-Fuller) is one of many common statistical tests for stationarity (in particular, unit root behavior) that may be used when fitting time series models to exchange rates (Mattera et al. 2021). The ADF sets the unit root (non-stationarity) hypothesis as the null (H_0).

If the ADF test show that the series are not stationary, using first differences can be useful. In that case, the differenced series can be expressed in the following way:

$$\Delta Y_t = Y_t - Y_{t-1}$$

Then, we need to identify the number of autoregressive and moving average terms, p and q, respectively, in order to construct the optimal ARIMA model to forecast exchange rates. Plots of the sample autocorrelation (ACF) and partial autocorrelation (PACF) functions are commonly used to identify the autoregressive term and moving average term. Partial autocorrelations are obtained by fitting a sequence of AR(k) models expressed in the following way:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_k Y_{t-k},$$

in which the last statistically significant lag value of the sample PACF determines the order of an AR model.

The Akaike Information Criterion (AIC) works by evaluating the model's fit on the training data and adding a penalty term for the complexity of the model. The desired result is to find the lowest possible AIC, which indicates the best balance between model fit and parsimony. This hopefully serves the eventual goal of getting the best possible forecast performance on the out-of-sample data (Brooks, 2014). It can be expressed in the following way:

$$AICc = -2\log(L) + 2(p + q + k + 1)\left[1 + \frac{(p + q + k + 2)}{T - p - q - k - 2}\right]$$

As shown in the formula, AIC uses the model's log-likelihood value as a measure of fit and trades-off with the number of fitted parameters. The model with the maximum likelihood is the one that "fits" the data the best. The natural log of the likelihood is used as a computational convenience.

Afterwards, the model fit is evaluated according to diagnostics tests to check if the residuals are approximately white noise. In particular, the residuals are required to have mean zero, constant variance and no autocorrelation (Seneviratna & Shuhua 2013). In addition, we would like the residuals to be normally distributed.

Finally, the optimal forecasts of the ARIMA processes are obtained as the expectation of $Y_{T+h}, h = 1,2, \dots$ conditional on present and past values of Y_T:

$$Y_{T+h|T} = E_T(Y_{T+h}) = E(Y_{T+h}|Y_T, Y_{T-1}, Y_{T-2}, \dots)$$

3.2 RNN and LSTM

A recurrent neural network (RNN) is a type of neural network specially designed to deal with time series. It can extract information of time series, allows information persistence, and use previous knowledge to infer follow-up patterns.

These deep learning algorithms are commonly used for ordinal or temporal problems, and like feedforward and convolutional neural networks (CNNs), recurrent neural networks utilize training data to learn. They differ from other algorithms mainly because of their "memory" as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depend on the prior elements within the sequence. While future events would also be helpful in determining the output of a given sequence, unidirectional recurrent neural networks cannot account for these events in their predictions. Figure 1 shows a visual representation of three types of RNNs.

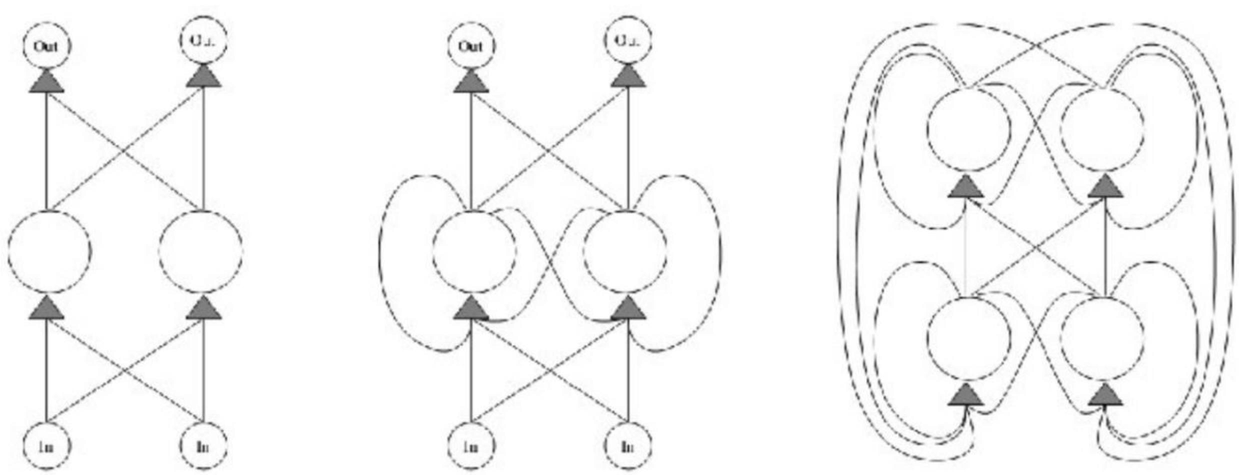

FIGURE 1 - Left, feed forward NN; Middle, simple recurrent neural network; Right, fully connected recurrent neural network.

The Long short-term memory (LSTM) model has been created to address some of the shortcomings of the RNN. A common LSTM consists of a certain number of cells. A cell consists of a forget gate an input gate and an output gate, as shown in Figure 2. The cell can remember values over time intervals. The input determines how much to add into current cell state from the input of current input and previous hidden state. The output gate determines the value of the next hidden state. The time constants associated with it are controlled by the forget gate (time t and cell i). The sigmoid unit sets the weight between 0 and 1.

$$f_i^{(t)} = \sigma(b_i^f + \sum_j U_{i,j}^f, x_j^{(t)} + \sum_j W_{i,j}^f, h_j^{(t-1)})$$

The current input vector is the current hidden layer vector. The output of LSTM cells is bias, input weight and circular weight of the forget gate. Therefore, the internal state of LSTM cells is updated in the following way, in which there is a condition of self-ring weight:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + \sigma\left(b_i + \sum_j U_{i,j}, x_j^{(t)} + \sum_j W_{i,j}, h_j^{(t-1)} \right)$$

Among them, the bias, input weight and cycle weight of forgetting gate in LSTM cells are calculated. The external input gate unit is updated in a way like the sigmoid but with its own parameters as shown below:

$$g_i^{(t)} = \sigma(b_i^g + \sum_j U_{i,j}^f, x_j^{(t)} + \sum_j W_{i,j}^g, h_j^{(t-1)})$$

The output gate of LSTM cells can also be closed (using sigmoid cells as gates) as expressed by:

$$q_i^{(t)} = \sigma(b_i^o + \sum_j U_{i,j}^o, x_j^{(t)} + \sum_j W_{i,j}^o, h_j^{(t-1)})$$

Cell status can be selected as additional input into the three gates of the first unit. Finally, the output from these three gates will combine and the neural network can decide which kinds of information need to be forget or reinforce. This memory property makes LSTM powerful and able to work with longer sequence data.

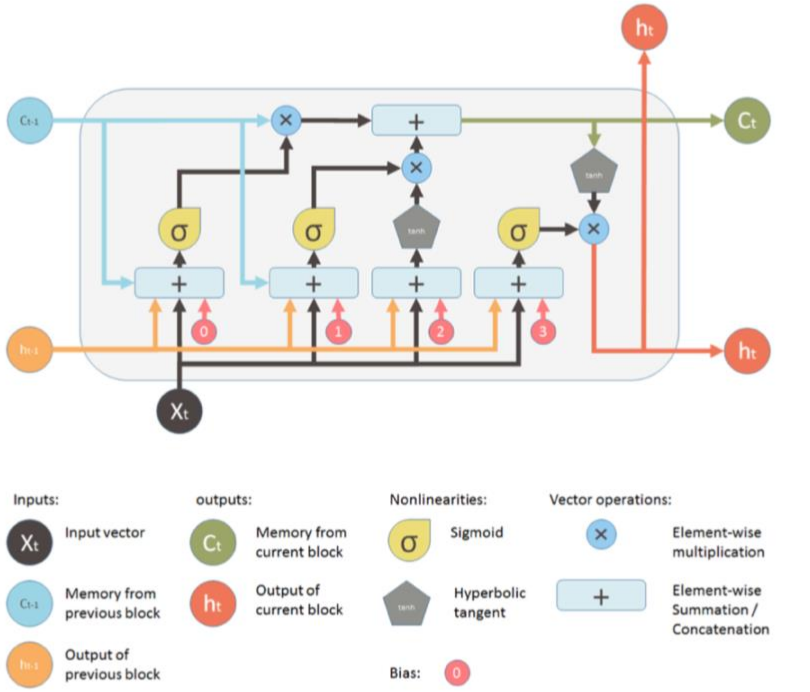

FIGURE 2 - Structure of the LSTM neural network. Reproduced from Yan, S.

3.2.1 Stochastic gradient descent

The goal of the stochastic gradient descent algorithm is to find a set of internal model parameters that will minimize a loss function such as logarithmic loss or mean squared error. The optimization algorithm is called 'gradient descent', where 'gradient' refers to the calculation of slope of error and 'descent' refers to the downwards path along that slope towards some minimum level of error. This is an iterative algorithm where in each step, the model is used with a set of internal parameters to make predictions on samples and compare them to real values, calculate the error, and update the internal parameters. The true gradient $\nabla Q(w)$ can be approximated by:

$$w := w - r\nabla Q(w),$$

where r is the learning rate.

The model of this paper is fitted using the efficient Adam version of stochastic gradient descent (Kingma, Ba 2015) to update network weights iterative based in training data. This optimization algorithm uses running averages of both the gradients and the second moments of the gradients. Adam's parameter update is given by:

$$m_w^{(t+1)} \leftarrow B_1 m_w^{(t)} + (1 - B_1)\nabla_w L^{(t)},$$

$$v_w^{(t+1)} \leftarrow B_2 v_w^{(t)} + (1 - B_2)\left(\nabla_w L^{(t)}\right)^2,$$

$$\widehat{m}_w = \frac{m_w^{(t+1)}}{1-B_1^t},$$

$$\hat{v}_w = \frac{v_w^{(t+1)}}{1 - B_2^t},$$

$$w^{(t+1)} \leftarrow w^{(t)} - n\frac{\widehat{m}_w}{\sqrt{\hat{v}_w + \varepsilon}},$$

given parameters $w^{(t)}$ and a loss function $L^{(t)}$ where t indexes the current training iteration. ε is a small scalar used to prevent division by zero, B_1 is the forgetting factor for gradients and B_2 the second moments of gradients.

3.2.2 Hyperparameter optimization

A procedure which is essential to the performance of the model is the selection of the hyperparameters. These refer to properties of the model that cannot be learned in the training process, therefore, they must be previously defined. However, the approaches usually used for fine-tuning these parameters are either computationally expensive (grid search) or of uncertain efficiency (trial-and-error), for this reason it is important to select a method based on the nature and complexity of the problem. The hyperparameters relevant to the LSTM model include but are not limited to batch size, hidden layers, number of units, dropout rate, learning rate, and number of epochs.

The batch size is a training sample of gradient descent. Each batch of samples will calculate a gradient descent to optimize the objective function. It can determine the direction of gradient descent and it is important to find the best balance between memory efficiency and memory capacity. When the batch size is properly increased, the number of epochs of the model will be reduced, as well as time expenditure, therefore, the accuracy of gradient descent direction will improve, and the amplitude of training vibration will be reduced.

The layers between the input and output layers are called hidden layers. There is no specific number on how many layers should be used, nonetheless, the selection of the number of hidden layers plays an important role in model prediction. If we choose a large number of layers, the model will perform extremely well in the training set, but rather poorly in the test set; this is a clear sign of overfitting. On the other hand, if the number of layers is too small, the model cannot extract data features very well, and its performance will be low.

The Learning Rate hyperparameter defines how quickly the network updates its parameters, in other words, it directly affects the convergence speed of the model to the local minimum. Setting a higher learning rate accelerates the learning but the model may not converge, or even diverge. On the contrary, a lower rate will slow down the learning drastically as steps towards the minimum of loss function will be minimal but will allow the model to converge smoothly.

The specified number of time steps defines the number of input variables (X) used to predict the next time step (y). This hyperparameter reflects how far back can the model remember. When the number of this parameter is small, the generalization ability of the model is better.

The dropout layer helps avoid overfitting in training by bypassing randomly selected neurons, therefore reducing the sensitivity to specific weights of the individual neurons. While adding more complexity such as increasing the number of nodes or dense layers, may risk overfitting, it can be addressed by adding a dropout rate to the LSTM layer.

The number of epochs defines the number times that the learning algorithm will work through the entire training dataset. An epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters and is comprised of one or more batches.

3.3 Random walk model

One of the simplest and yet most important models in time series forecasting is the random walk (RW) model. This model assumes that in each period the variable takes a random step away from its previous value, and these steps are independently and identically distributed in size. The RW model can be with drift or without drift depending on whether the distribution of step sizes has a non-zero mean or a zero mean. The forecasts of the driftless random walk (DRW) can be expressed in the following way:

$$Y_{T+h|T} = Y_T$$

which means that, at period T, the h-step-ahead forecast of variable Y for the random walk is the observed value at that period, Y_T. Thus, the model will predict that all future values are equal to the last observed value. In other words, the probability of the series of taking an upwards or downwards step is the same. For the RW with drift model, the h step-ahead forecast from period n is expressed as:

$$\hat{Y}_{T+h|T} = Y_T + h\hat{d}$$

Where the term $\hat{d}$ is the drift, which can be estimated using the sample average change of ΔY from $\mathrm{t} = 1, \dots, \mathrm{T}$.

In the case of exchange rates, there is no evidence to assume a long-term trend in one direction or the other. Due to market efficiency, exchange rates are largely viewed to be best explained as Random Walk models without drift (Diebold & Nason, 1990). The weak form of the Efficient Markets Hypothesis $(EMH)^2$ states that, in an efficient market, asset prices fully reflect all available information about the asset, thus, investors cannot consistently earn abnormal returns (Peirson et al., 1995).

In addition, the previously mentioned research by Meese and Rogoff (1983), and Hogan (1986) supports the superiority of Random Walk over certain alternative forecasting models.

3.4 Model evaluation

Due to the sufficiently large dataset available (5,777 datapoints) and to its computational efficiency, a simple train-test split was chosen to evaluate the models developed in this study.

This procedure involves taking a dataset and dividing it into two subsets. The first subset (training set) is used to fit the model whereas the second subset (test set) is used to evaluate the fit of the model. The input element of the dataset is provided to the model then predictions are made and compared to the expected values.

The data used in this paper was split in the following way. First, the length of the time horizons was taken out of the dataset; for the first time-horizon 5,772 data points were used to feed the models and the last 5 data points were reserved to evaluate its accuracy. For the second time-horizon, 5,717 data points were fed into the models, and 60 data points were reserved for evaluation. The reserved data points were considered as 'out-of-sample'. This split was made considering that the data set is expressed in business days, therefore, a forecast of one business week (5 days) and three business months (60 days) were made.

Finally, to fit and train each model and adjust the parameters to improve the accuracy of the forecasts, the data was split into 80% for the training set and 20% for the test set. Figure 3 illustrates the data partitioning scheme used in this paper.

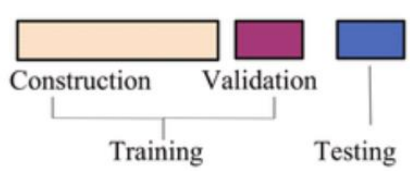

FIGURE 3 - Single train-test split with an external testing set. Reproduced from Chen, Y.

3.5 Accuracy metrics

Two measures were selected to calculate the forecasting errors of the models in this investigation: MAPE and the MSE.

The mean absolute percentage error (MAPE) is obtained as the average of the percentage forecast errors and it can be expressed mathematically in the following way:

$$MAPE = mean(\left| {}^{e_t}/_{Y_t} \right|)$$

where e_t is the forecast error, *i.e.*, the difference between the observed value at period t, Y_t, and its corresponding forecast.

The mean-square error (MSE) is obtained as the average of the forecast errors squared and can be written mathematically as:

$$MSE = mean(e_t^2)$$

4. Data

4.1. Source

The dataset used in this paper was retrieved from the United States Board of Governors of the Federal Reserve System download data program, and it is publicly available through their website (https://www.federalreserve.gov/datadownload). The parameters of the data such as start date, end date, and frequency can be modified before the extraction.

4.2. Data processing

The dataset consists of exchange rates for the last 22 years, from January 1999 to December 2021, of the currency pair United States Dollar/Euro. The frequency of the data is business days. By looking at the plot presented in Figure 4, we can clearly observe the fluctuation of the USD over time, reaching its highest value against the euro in the early 2000s (1 euro = 0.827 USD) and later drastically losing value until reaching its lowest point in 2008 (1 euro = 1.601 USD) due to the well-known financial crisis caused by the collapse of the housing market in the United States. Table 1 shows a summary of the numerical characteristics of this dataset.

USD/EUR exchange rate dataset	
count	5770
mean	1.1992
std	0.1595
min	0.8270
25%	1.1026
50%	1.1983
75%	1.3168
max	1.6010

TABLE 1: Description of the dataset

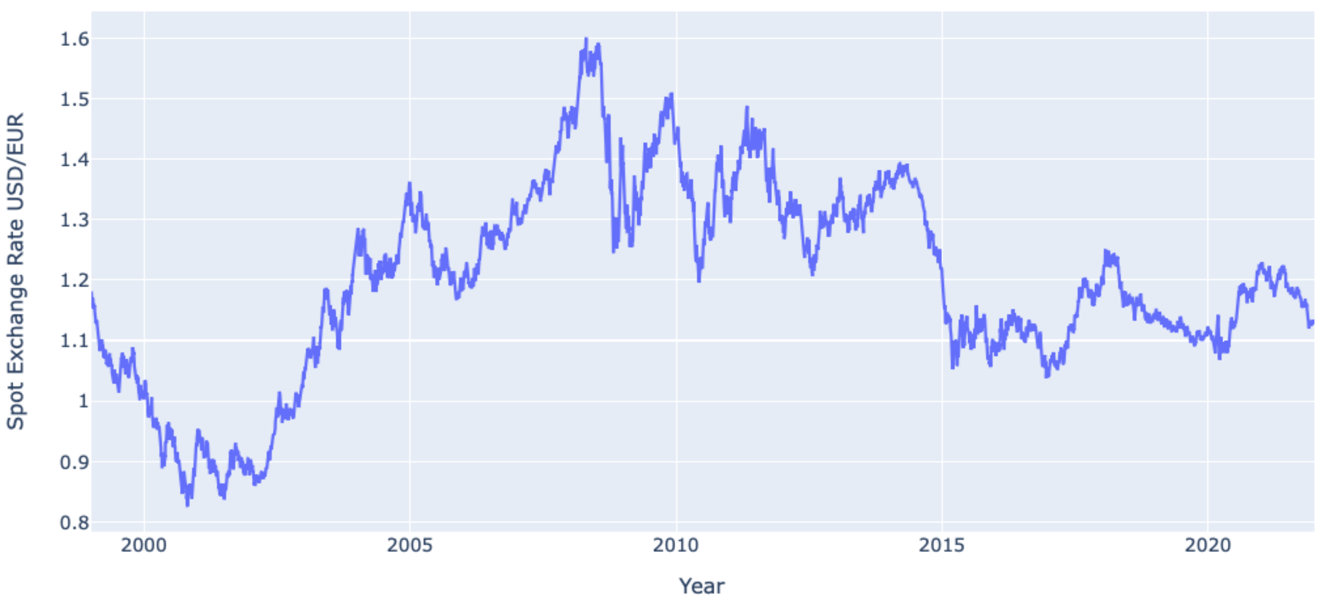

FIGURE 4: Timeseries for the currency pair USD/EUR. Each datapoint represents the daily value of 1 USD compared to 1 Euro.

5. Analysis of experimental results

5.1 ARIMA

The ARIMA model was developed using the Box-Jenkins procedure. First, I performed an ADF test to determine if the data was stationary. The results of the original series couldn't reject the H_0 that the time series were stationary, therefore, first difference series were used ($\Delta Y_t = Y_t - Y_{t-1}$) and the ADF test was performed again, this time, rejecting H_0 and confirming stationarity.

Second difference series were also calculated for comparison purposes (Table 2) but for the actual model I used the term $(d) = 1$ in the ARIMA (p, d, q) structure. Figure 5 shows a plot of the first difference series.

	Original series	First difference series	Second difference series
Test Statistic:	-1.68444637	-75.10178149	-21.66483649
p-value:	0.439143463	0.00E+00	0.00E+00
Significance Levels:			
1%:	-3.431	-3.431	-3.431
5%:	-2.862	-2.862	-2.862
10%:	-2.567	-2.567	-2.567

TABLE 2: Results of the Augmented Dickey-Fuller Test

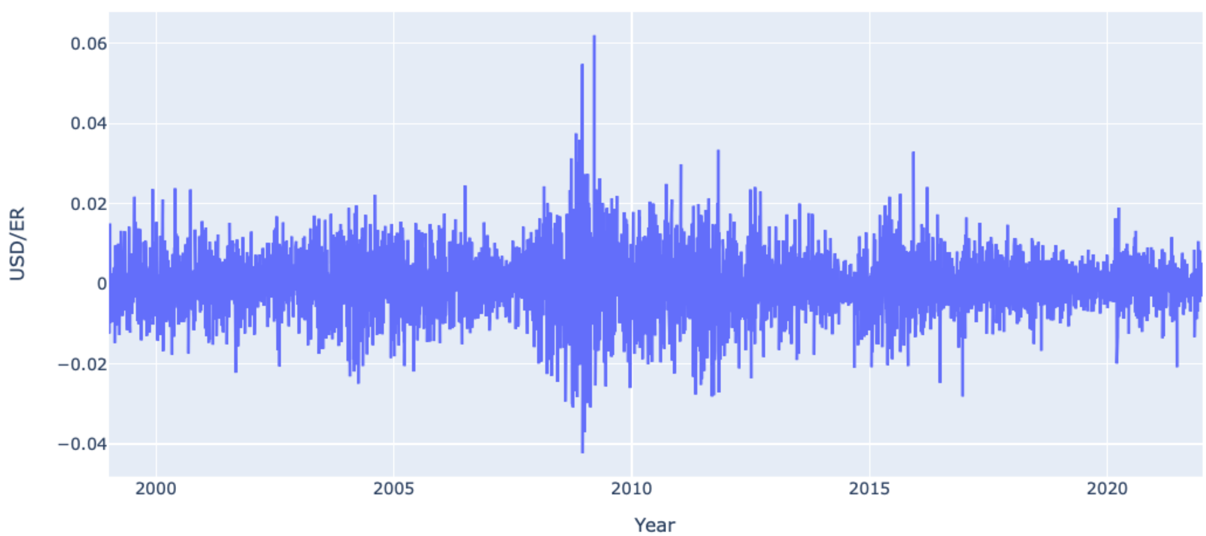

FIGURE 5: Plot of the first difference series.

Next, I produced plots of the ACF and PACF (Figure 6 and 7) of the original and the first difference series in order to determine the autoregressive term and moving average term. These plots show peaks at lag 0, suggesting an ARIMA (0, 1, 0) which is consistent with a random walk model. However, since these plots seem inconclusive and, to be certain, I produced an iterative AIC test that tried several combinations of the *p* and *q* terms with the *d* term fixed at 1. The results of the test showed that the (2, 1, 2) model presented the lowest AIC (Table 3) and therefore, the parameters were selected accordingly ($p = 2$; $q = 2$).

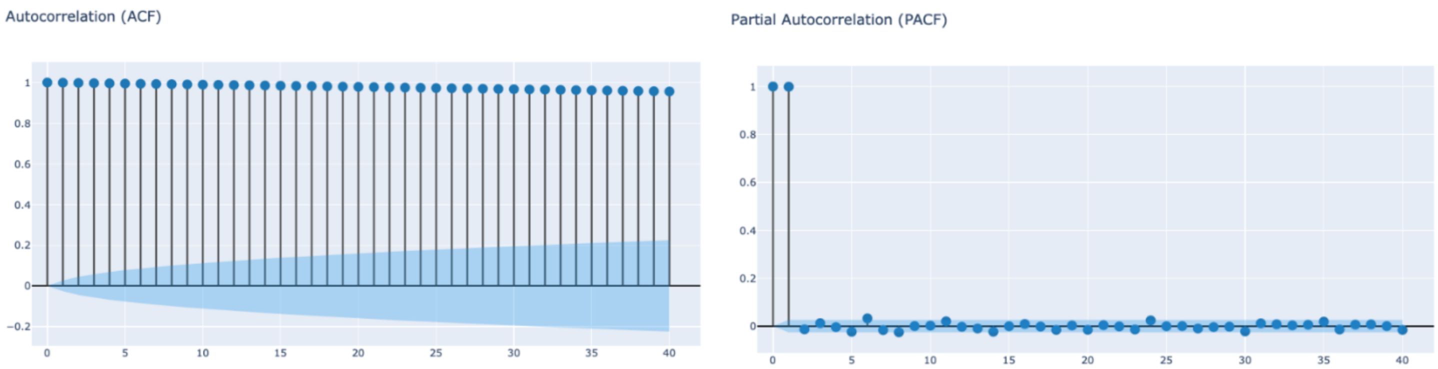

FIGURE 6: Plot of the ACF and PACF for the original series, using a sample of 40 lags

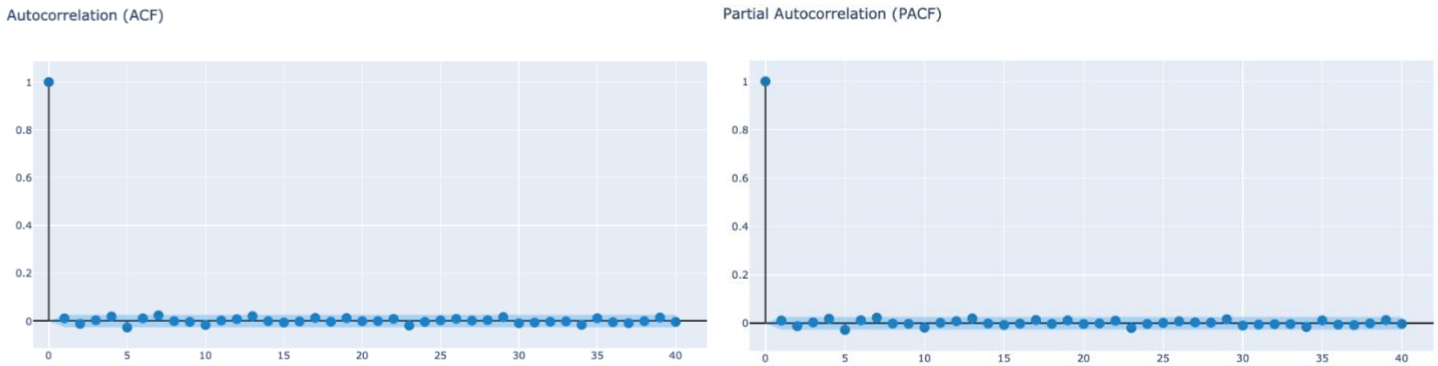

FIGURE 7: Plot of the ACF and PACF for the first difference series, using a sample of 40 lags

AIC Result	Model
-40586.29898	(2, 1, 2)
-40584.91414	(3, 1, 2)
-40584.90336	(2, 1, 3)
-40584.21098	(1, 1, 1)
-40584.08689	(4, 1, 2)
-40583.28875	(4, 1, 1)
-40582.67430	(3, 1, 3)
-40582.22436	(1, 1, 2)
-40581.85149	(2, 1, 1)
-40581.38558	(4, 1, 3)
-40581.37007	(3, 1, 1)
-40581.29421	(2, 1, 1)

TABLE 3: Results of the Akaike Information Criterion for different ARIMA models

The next step was to fit the model and analyze the residual errors to see if these are random and, ideally, normally distributed. In this step, the data was split into 80%/20% for the train and test set respectively and the ARIMA module from the statsmodels library was used. Figure 8 shows the plot of the residual errors on the left, and the density of the errors on the right. From these, it can be clearly observed that the residuals errors of the model are randomly distributed around zero and follow a normal distribution. Additionally, an analysis of the residuals is presented in Table 4 which show a mean value very close to zero, therefore, our diagnostic tests conclude that the chosen model is an appropriate fit for the data. The detailed results of the ARIMA(2,1,2) are presented in Table 7 in the Annex.

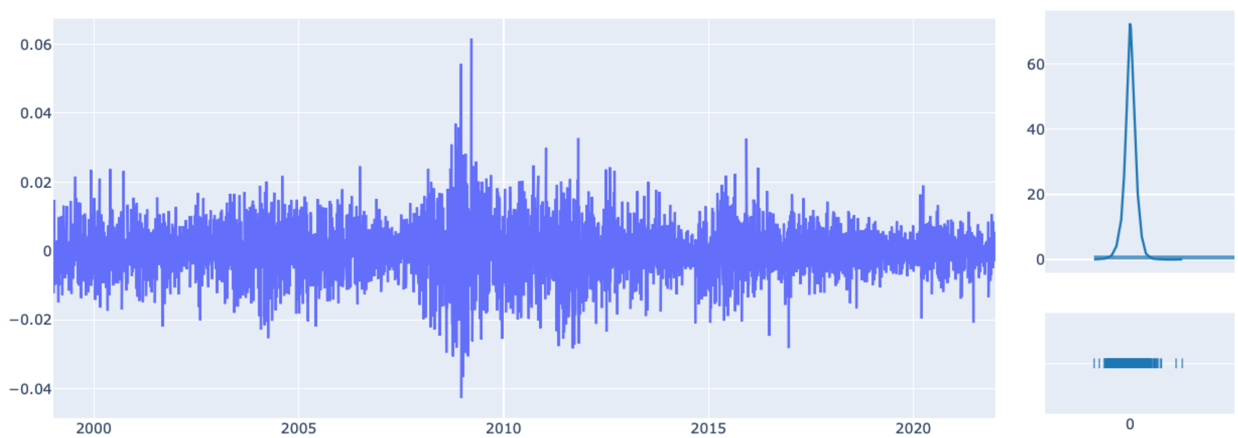

FIGURE 8: Plots of the residual errors for the ARIMA model (2, 1, 2)

Description of the data	
count	5769
mean	2.47E-08
std	7.17E-03
min	-4.27E-02
25%	-3.69E-03
50%	-1.71E-05
75%	3.89E-03
max	6.16E-02

TABLE 4: Description of the residuals of the fitted ARIMA model

Finally, the dataset was split as shown in figure 3, with a test set of 5 business days for the 5-day horizon and 60 business days for the 60-day horizon respectively and the ARIMA (2,1,2) was fitted in both instances of the dataset. The resulting predicted data points ($\hat{Y}$) were used in the selected error metrics to evaluate its performance. The results are presented in Tables 5 and 6 of section 5.4

5.2 LSTM

Similar to ARIMA, the first step to develop the LSTM model is to check for stationarity in the data because a non-stationary series will introduce more error in predictions and force errors to compound faster. Since the ADF tests were previously performed, the first difference of the data ($\Delta Y_t = Y_t - Y_{t-1}$) was used to fit this model.

The data format required for the LSTM is three-dimensional, so in order to feed it to the model it is necessary to split it into samples with an input and output component, and later reshape it to structure of samples, timesteps, and features. The determination of the features component is the most straightforward; I am using a univariate time series; therefore, it was set to 1. Additionally, a window size of 5 days and 60 days was used as a lookback to produce the 5-day and 60-day forecasts respectively. The data was split in the same fashion as the ARIMA model, described in Figure 3.

LSTM models are sensitive to the scale of the input data, specifically when the sigmoid or tanh activation functions are used. To address this, the data was normalized using a min-max scaler function, which can be expressed in the following way:

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Afterwards, I created a learning rate scheduler. This scheduler monitored the validation loss and modified the learning rate on plateaus. The mean squared error (MSE) was chosen as the loss function of the LSTM model, and the threshold to modify the learning rate was set to 0.1 with a patience of 100 epochs.

Finally, I created 2 LSTM hidden layers, and an output dense layer with a sigmoid activation function, using the tensorflow keras sequential library. The hyperparameters of the model were chosen using a trial-and-error methodology, testing several combinations of hyperparameters, and comparing the results of the models with the error metrics described in section 3.4. The hyperparameters include batch size, hidden layers, number of units, dropout rate, learning rate, and number of epochs. Table 8 in the Annex shows the selected settings of the hyperparameters used in this study.

The LSTM model was fitted using the Adam version of stochastic gradient descent to update network weights iterative based in training data. Figure 9 shows the model loss over epoch for the 5-day horizon, in which it can be observed that the model converges rather quickly for both train and test sets.

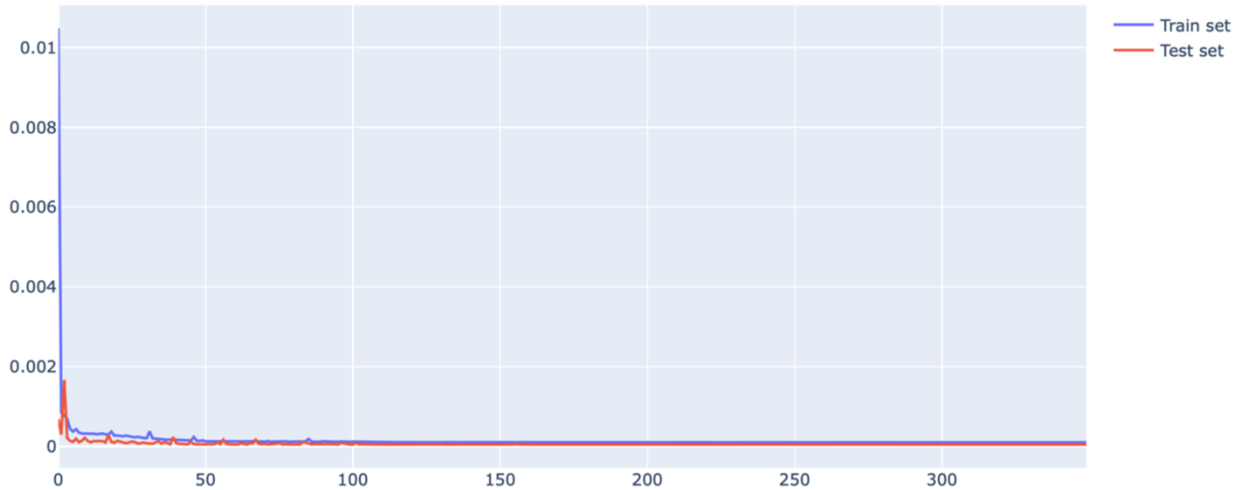

FIGURE 9: Model loss over epoch for the 5-day horizon

The results of the LSTM model in both horizons are presented in Tables 5 and 6 of section 5.4, and the predicted values in Tables 9 and 10 in the Annex.

5.3 Driftless Random Walk

To produce the forecast of this model, the simple formula $\hat{Y}_{n+k} = Y_n$ was used. In other words, I took the value of the last lag and reproduced it for the next 5 and 60 lags for the 5-day and 60-day horizons respectively. Finally, the errors were calculated using the metrics described in section 3.5. The results of this model are presented in Tables 5 and 6 of section 5.4, and the predicted values in Tables 9 and 10 in the Annex.

5.4 Results

The predictive power of the ARIMA (2, 1, 2) model, LSTM neural network model, and the driftless random walk was compared in a 5-day and 60-day horizon for the currency pair USD/EUR. The data was split into train and test set to calculate the accuracy of the models as described in section 3.4. Additionally, the mean absolute percentage error (MAPE), root mean squared error (RMSE), and mean squared error (MSE) were used as metrics to calculate the errors of the models as described in section 3.5. Table 5 and 6 show the out-of-sample accuracy measures for the three models in both time horizons.

5.4.1 Five-day horizon

The DRW model had the best overall performance when predicting exchange rates for 5 business days ahead, thus adding evidence of the difficulty of forecasting exchange rates at short horizons and of outperforming a RW benchmark. Despite of this, the difference in the error metrics between the ARIMA and the DRW models is minimal (~2% in MSE) and shows that the former is a perfectly suitable model for exchange rates. The LSTM model showed the highest error in this horizon; nevertheless, the out-of-sample performance of this model is still good and perhaps a fine-tuning of its hyperparameters via grid search can make a significant difference in its results. Table 5 shows the error metrics for the three models in the 5-day horizon, whereas figures 10 and 11 show a plot of the forecasted points of the ARIMA and LSTM models respectively.

	ARIMA (2,1,2)	LSTM	DRW
Mean Absolute Percentage Error (MAPE)	7.51E-02	2.90E-01	7.60E-02
Mean Squared Error (MSE)	7.03E-07	9.18E-06	6.92E-07

TABLE 5: Results of the error metrics for ARIMA, LSTM, and DRW models on a 5-day horizon.

FIGURE 10: plot of the ARIMA(2,1,2) forecast in the 5-day horizon. The training set was reduced for visualization purposes.

FIGURE 11: plot of LSTM forecast in the 5-day horizon. The training set was reduced for visualization purposes.

5.4.2 Sixty-day horizon

The models were also tested when forecasting 60 timesteps into the future. In this case, all three models showed a slight deterioration of their predicting accuracy compared to the 5-day horizon. Nevertheless, both the LSTM and the ARIMA(2,1,2) outperformed the DRW benchmark, the former showing the lowest error with a MAPE of 1.71E+00 and a MSE of 4.60E-04 and a difference of MSE with the DRW of ~6%. The ARIMA model shows a MAPE of 1.73E+00 and a MSE of 4.82E-04. These results show that both autoregressive and machine learning models could outperform the DRW benchmark and should be widely considered when forecasting univariate exchange rate series. Table 6 shows the error metrics for the three models in the 60-day horizon, while figures 12 and 13 show a plot of the forecasted points of the ARIMA and LSTM models respectively.

	ARIMA (2,1,2)	LSTM	DRW
Mean Absolute Percentage Error (MAPE)	1.73E+00	1.71E+00	1.74E+00
Mean Squared Error (MSE)	4.82E-04	4.60E-04	4.88E-04

TABLE 6: Results of the error metrics for ARIMA, LSTM, and DRW models on a 60-day horizon.

FIGURE 12: plot of the ARIMA(2,1,2) forecast in the 60-day horizon. The training set was reduced for visualization purposes

FIGURE 13: plot of the LSTM forecast in the 60-day horizon. The training set was reduced for visualization purposes.

6. Conclusion

This study applied autoregressive (ARIMA) and machine learning (LSTM) models to predict exchange rates of the USD/EUR currency pair for a 5-day and 60-day horizon and compared both models to the driftless random walk benchmark to measure their accuracy.

Exchange Rate forecasting has been longstanding problem for economists and data analysts due to its difficulty and its unpredictable nature which has led several investigators to conclude that exchange rates are largely disconnected from economic fundamentals. Previous studies and new developments in computing power and a widespread availability of data have triggered the development of new and hybrid models. New models derived from machine learning algorithms have presented promising results, in some cases outperforming the RW benchmark. Hybrid models using economic fundamentals with machine learning models have also produced accurate predictions and suggest that the collaboration of traditional disciplines like Economics with Data Analytics can improve significantly the accuracy of model outputs, can reduce the complexity of variable selection, or of hyperparameter selection, to name a few examples.

By using publicly available data from the United States Board of Governors of the Federal Reserve System download data program, this study has shown the predictive power of both the ARIMA and the LSTM models and demonstrated that the univariate LSTM model outperformed the driftless random walk benchmark in the 60-day horizon, thus adding evidence that machine learning models can present increased accuracy in medium horizons. On the other hand, the DRW benchmark had better accuracy than the other two models in the 5-day horizon, which is consistent with the existing literature, showing once more the difficulty of forecasting exchange rates at short horizons. Even so, both the ARIMA and the LSTM models achieved promising results with very low errors, and further tests should be performed, perhaps with a more extensive dataset, to test their accuracy.

References

Ariyo, A., Adewumi, A., & Ayo, C. (2014). Stock price prediction using the arima model. *UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 106–112.

Abedin, M.Z., Moon, M.H., Hassan, M.K., & Hajék, P. (2021). Deep learning-based exchange rate prediction during the COVID-19 pandemic. *Annals of Operations Research*, 1-52.

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

Adebiyi, A. A., Ayo, C. K., Adebiyi, M., & Otokiti, S. O. (2012). Stock price prediction using neural network with hybridized market indicators. *Journal of Emerging Trends in Computing and Information Sciences*, 1-9.

Babu, A. S., & Reddy, S. K. (2015). Exchange rate forecasting using ARIMA, Neural Network and Fuzzy Neuron. *Journal of Stock & Forex Trading*, 1-5.

Bajari, P., Nekipelov, D., Stephen P., & Yang, M. (2015). Machine Learning Methods for Demand Estimation. *American Economic Review*, 481-85.

Berben, R., & Van Dijk, D. (1998), Does the Absence of Cointegration Explain the Typical Findings in Long Horizon Regressions? *Erasmus University Econometrics Institute*, (No. 9814).

Berkowitz, J., & Giorgianni, L. (2001). Long-Horizon Exchange Rate Predictability? *Review of Economics and Statistics*, 81-91.

Bilson, J.F. (1978). The Monetary Approach to the Exchange Rate: Some Empirical Evidence. *International Monetary Fund Staff Papers*, 48-75.

Brooks, C. (2014). Introductory econometrics for finance. *Cambridge Books*.

Carriero, A., Kapetanios, G., & Marcellino, M. (2009). Forecasting exchange rates with a large Bayesian VAR. *International Journal of Forecasting*, 400-417.

Chattopadhyay, G., & Chattopadhyay, S. (2012). Monthly sunspot number time series analysis and its modeling through autoregressive artificial neural network. *The European Physical Journal Plus*, 1-8.

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

Collantes-duarte, J., & Rivas-echeverría, F. (2002). Time Series Forecasting using ARIMA, Neural Networks and Neo Fuzzy Neurons. In *3rd WSEAS International Conference on Neural Networks and Applications*, 11-14.

Diebold, F., & Nason, J. (1990). Nonparametric exchange rate prediction*? Journal of International Economics*, 315-332.

Dornbusch, R. (1976). Expectations and Exchange Rate Dynamics. *Journal of Political Economy*, 1161–1176.

Engel, C., & Hamilton, J. (1990). Long Swings in the Dollar: Are They in the Data and Do Markets Know It? *American Economic Review*, 689-713.

Engel, C., Lee, D., Liu, C., Liu, C., & Wu, S. (2018). The Uncovered Interest Parity Puzzle, Exchange Rate Forecasting, and Taylor Rules. *Journal of International Money and Finance*, 317-331.

Engel, C., Nelson, M., & Kenneth, D. (2007). Exchange rate models are not as bad as you think. *NBER Macroeconomics Annual Review*, 381-473.

Faust, J., Rogers, J., & Wright, J. (2003). Exchange Rate Forecasting: the Errors We've Really Made. *Journal of International Economics*, 35-59.

Fleming, J. M. (1962). Domestic Financial Policies under Fixed and under Floating Exchange Rates. *International Monetary Fund Staff Papers*, 369–380.

Frankel, J. A. (1979). On the Mark: A Theory of Floating Exchange Rates Based on Real Interest Differentials. *The American Economic Review*, 610–622.

Gorgolis, H., & Grad, G. (2019). Hyperparameter Optimization of LSTM Network Models through Genetic Algorithm. *10th International Conference on Information, Intelligence, Systems and Applications*, 1-4.

Granger, C. W., & Terasvirta, T. (1993). Modelling Non-linear Economic Relationships. *Southern Economic Journal*, 1241–1243.

Groen, J.J. (1999). Long Horizon Predictability of Exchange Rates: is it for Real? *Empirical Economics*, 451-469.

Haggan, V., & Ozaki, T. (1981). Modelling Nonlinear Random Vibrations Using an Amplitude-Dependent Autoregressive Time Series Model. *Biometrika*, 189-96.

Hawley, D.D., Johnston, J.D., & Raina, D. (1993). Artificial neural systems: A new tool for financial decision making. *Financial Analysts Journal,* 63–72.

Hogan, L. (1986). A Comparison of Alternative Exchange Rate Forecasting Models. *Economic Record,* 215-223.

Hsieh, D. (1988). The statistical properties of daily foreign exchange rates: 1974-1983. *Journal of International Economics,* 129-145.

Irena M., & Andrius B. (2013). Fundamental Exchange Rate Forecasting Models: Advantages and Drawbacks. *KSI Transactions on Knowledge Society*, 6(3).

Kamruzzaman, J., & Sarker R. (2004). ANN-Based Forecasting of Foreign Currency Exchange Rates. *Neural Information Processing - Letters and Reviews,* 49-58.

Karemera, D., & Kim, B. J. (2006). Assessing the forecasting accuracy of alternative nominal exchange rate models: the case of long memory. *Journal of Forecasting,* 369-380.

Kaynar, O., & Taştan, S. (2009). Zaman Serisianalizinde MLP Yapay Sinir Ağları Ve Arıma Modelinin Karşılaştırılması. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 161–172.

Kilian, L. (1999). Exchange Rates and Monetary Fundamentals: What Do We Learn From Long-Horizon Regressions? *Journal of Applied Econometrics, 491-510.*

Kingma, D., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.

Lee, J., & Jhee, W. (1994). A two stage neural network approach for ARMA model identification with ASACF. *Decision Support Systems*, 461-479.

Lucas, R. (1982). Interest Rates and Currency Prices in a Two-Country World. *Journal of Monetary Economics*, 335-359.

Mark, N. (1995). Exchange rates and fundamentals: Evidence on long-horizon predictability. *American Economic Review,* 201-18.

Meese, R., & Kenneth, R. (1983a). Empirical exchange rate models of the seventies: Do they fit out of sample? *Journal of International Economics,* 3-24.

______.(1983b). The out-of-sample failure of empirical exchange rate models: Sampling error or misspecification. In *Exchange rates and international macroeconomics, University of Chicago Press*, 67-105.

Mehta, M. (1995). Foreign Exchange Markets. *Neural Networks in the Capital Markets*, 177-198.

Michael, P., Nobay, A., & Peel, D. (1997). Transactions Costs and Nonlinear Adjustment in Real Exchange Rates; An Empirical Investigation. *Journal of Political Economy*, 862–879.

Mundell, R. A. (1968). International economics. *Macmillan*.

Nwankwo, S. C. (2014). Autoregressive Integrated Moving Average (ARIMA) Model for Exchange Rate (Naira to Dollar). *Academic Journal of Interdisciplinary Studies*, 429.

Olajide, J. T., Ayansola, O., Odusina, M., & Oyenuga, I. F. (2012). Forecasting the Inflation Rate in Nigeria: Box Jenkins Approach. *IOSR Journal of Mathematics (IOSR-JM)*, 5.

Özkan, F. (2011). Döviz Kuru Tahmininde Yapay Sinir Ağlarıyla Alternatif Yaklaşım.", *Eskişehir Osmangazi Üniversitesi, İİBF Dergisi*, 2.

Peirson. (1995). Business finance (6th ed.), *McGraw-Hill.*

Pfahler, J. (2021). Exchange Rate Forecasting with Advanced Machine Learning Methods. *Journal of Risk and Financial Management,* 1-17.

Pincheira, P., & Neumann, F. (2018). Can We Beat the Random Walk? The Case of Survey-Based Exchange Rate Forecasts in Chile. *Finance Research Letter,* 37.

Plakandaras, V., Papadimitriou, T., & Gogas, P. (2017). Forecasting Daily and Monthly Exchange Rates with Machine Learning Techniques. *Journal of Forecasting*, 560-573.

Qu, Y., & Zhao, X. (2019). Application of LSTM Neural Network in Forecasting Foreign Exchange Price. In *Journal of Physics*: *Conference Series*, 1237(4).

Mattera, R., Misuraca, M., Scepi, G., & Spano, M. (2021). A mixed-frequency approach for exchange rates prediction. *University of Salento.*

Refenes, A. (1992). Constructive learning and its applications to currency exchange rate forecasting. *Neural Networks in Finance and Investing*, 465-494.

Rossi, B. (2013). Are Exchange Rates Predictable? *Journal of Economic Literature*, 1063-1119.

Rout, M., Majhi, B., Majhi, R., & Panda, G. (2014). Forecasting of currency exchange rates using an adaptive ARMA model with differential evolution-based training. *Journal of King Saud University – Computer and Information Sciences*, 7-18.

Seneviratna, D., & Shuhua, M. (2013). Forecasting the Twelve-Month Treasury Bill Rates in Sri Lanka: Box Jenkins Approach. *IOSR Journal of Economics and Finance (IOSR-JEF)*, 1(1).

Stockman, A.C. (1980). A Theory of Exchange Rate Determination. *Journal of Political Economy*, 673-698.

Teräsvirta, T. (1994). Specification, estimation, and evaluation of smooth transition autoregressive models. *Journal of the american Statistical association*, 208-218.

Vitek, F. (2005). The Exchange Rate Forecasting Puzzle. *International Monetary Fund Staff Papers, 47.*

Weisang, G., & Awazu, Y. (2014). Vagaries of the Euro: An Introduction to ARIMA Modeling. *Case Studies in Business, Industry and Government Statistics*, 45-55.

Yao, J., & Tan C. (2000). A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*, 79-98.

Yıldırım, D., Toroslu, I., & Fiore, U. (2021). Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators. *Southwestern University of Finance and Economics*, 1-36.

Zhang, Y., & Wu, H. (2007). A comparison of the prediction performances by the linear models and the ARIMA model. *UMEA School of Business*.

Zhang, X. (1994). Non-linear predictive models for intra-day foreign exchange trading. *Intelligent Systems in Accounting, Finance and Management*, 293-302.

Websites which were very useful for this work:

Brownlee, J. (2018). *Difference Between a Batch and an Epoch in a Neural Network.* Retrieved from: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/

Brownlee, J. (2018). *How to Develop LSTM Models for Time Series Forecasting*. Retrieved from: https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

Mushailov, J. (2021). *LSTM Framework for Univariate Time-Series Prediction*. Retrieved from: https://towardsdatascience.com/lstm-framework-for-univariate-time-series-prediction-d9e7252699e

Nguyen, A. (2020). *Exchange Rate Prediction: Time Series Forecasting with ARIMA*. Retrieved from: https://towardsdatascience.com/exchange-rate-prediction-time-series-forecasting-with-arima-27260faafcdc

Prabhakaran, S. (2021). *ARIMA Model – Complete Guide to Time Series Forecasting in Python*. Retrieved from: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

Yan, S. (2016). *Understanding LSTM and Its Diagrams.* Retrieved from: https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714

Annexes

```
ARIMA Model Results
==============================================================================
Dep. Variable:              D.USD/EUR   No. Observations:                 5769
Model:                 ARIMA(2, 1, 2)   Log Likelihood               20299.149
Method:                       css-mle   S.D. of innovations              0.007
Date:                Tue, 15 Mar 2022   AIC                         -40586.299
Time:                        00:36:33   BIC                         -40546.337
Sample:                             1   HQIC                        -40572.394

===================================================================================
                      coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------
const           -8.578e-06   9.54e-05     -0.090      0.928      -0.000       0.000
ar.L1.D.USD/EUR    -1.4331      0.192     -7.447      0.000      -1.810      -1.056
ar.L2.D.USD/EUR    -0.8508      0.187     -4.549      0.000      -1.217      -0.484
ma.L1.D.USD/EUR     1.4498      0.181      8.010      0.000       1.095       1.805
ma.L2.D.USD/EUR     0.8688      0.178      4.892      0.000       0.521       1.217
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -0.8423           -0.6827j            1.0842           -0.3916
AR.2           -0.8423           +0.6827j            1.0842            0.3916
MA.1           -0.8344           -0.6744j            1.0729           -0.3918
MA.2           -0.8344           +0.6744j            1.0729            0.3918
-----------------------------------------------------------------------------
```

TABLE 7 - Detailed results of the ARIMA(2,1,2) model

Hyperparameter	Chosen Value
Batch Size (5-day; 60-day)	1154; 96
Hidden Layers	2
Units per layer	100
Dropout Rate	0.4
Learning Rate	0.001
Number of Epochs	350

TABLE 8 - Chosen hyperparameters for the LSTM model

Date	y	yhat		
		LSTM	ARIMA	DRW
23/12/21	1.1320	1.1297	1.1323	1.1324
27/12/21	1.1329	1.1296	1.1323	1.1324
28/12/21	1.1314	1.1296	1.1324	1.1324
29/12/21	1.1337	1.1295	1.1323	1.1324
30/12/21	1.1318	1.1294	1.1323	1.1324

TABLE 9 – Predicted values of the LSTM, ARIMA, and DRW models in the 5-day horizon

Date	y	yhat		
		LSTM	ARIMA	DRW
4/10/21	1.1622	1.1613	1.1599	1.1598
5/10/21	1.1609	1.1615	1.1597	1.1598
6/10/21	1.1546	1.1617	1.1598	1.1598
7/10/21	1.1561	1.1618	1.1598	1.1598
8/10/21	1.1572	1.1617	1.1597	1.1598
12/10/21	1.1541	1.1616	1.1597	1.1598
13/10/21	1.1568	1.1615	1.1598	1.1598
14/10/21	1.1591	1.1614	1.1597	1.1598
15/10/21	1.1594	1.1614	1.1597	1.1598
18/10/21	1.1609	1.1613	1.1598	1.1598
19/10/21	1.1632	1.1612	1.1597	1.1598
20/10/21	1.1643	1.1611	1.1597	1.1598
21/10/21	1.1643	1.1610	1.1598	1.1598
22/10/21	1.1632	1.1609	1.1597	1.1598
25/10/21	1.1609	1.1608	1.1597	1.1598
26/10/21	1.1590	1.1608	1.1597	1.1598
27/10/21	1.1600	1.1607	1.1597	1.1598
28/10/21	1.1685	1.1606	1.1597	1.1598
29/10/21	1.1552	1.1606	1.1597	1.1598
1/11/21	1.1591	1.1605	1.1597	1.1598
2/11/21	1.1581	1.1605	1.1597	1.1598
3/11/21	1.1584	1.1605	1.1597	1.1598
4/11/21	1.1546	1.1604	1.1597	1.1598
5/11/21	1.1554	1.1604	1.1597	1.1598

8/11/21	1.1590	1.1603	1.1597	1.1598
9/11/21	1.1589	1.1603	1.1597	1.1598
10/11/21	1.1517	1.1602	1.1597	1.1598
12/11/21	1.1443	1.1601	1.1597	1.1598
15/11/21	1.1421	1.1601	1.1597	1.1598
16/11/21	1.1333	1.1600	1.1597	1.1598
17/11/21	1.1322	1.1599	1.1597	1.1598
18/11/21	1.1358	1.1598	1.1597	1.1598
19/11/21	1.1318	1.1598	1.1596	1.1598
22/11/21	1.1260	1.1597	1.1596	1.1598
23/11/21	1.1265	1.1596	1.1596	1.1598
24/11/21	1.1196	1.1595	1.1596	1.1598
26/11/21	1.1302	1.1594	1.1596	1.1598
29/11/21	1.1261	1.1594	1.1596	1.1598
30/11/21	1.1287	1.1593	1.1596	1.1598
1/12/21	1.1323	1.1592	1.1596	1.1598
2/12/21	1.1306	1.1591	1.1596	1.1598
3/12/21	1.1308	1.1591	1.1596	1.1598
6/12/21	1.1282	1.1590	1.1596	1.1598
7/12/21	1.1247	1.1589	1.1596	1.1598
8/12/21	1.1330	1.1588	1.1596	1.1598
9/12/21	1.1285	1.1587	1.1596	1.1598
10/12/21	1.1312	1.1587	1.1596	1.1598
13/12/21	1.1298	1.1586	1.1596	1.1598
14/12/21	1.1267	1.1585	1.1596	1.1598
15/12/21	1.1261	1.1584	1.1596	1.1598
16/12/21	1.1309	1.1583	1.1596	1.1598
17/12/21	1.1277	1.1583	1.1596	1.1598
20/12/21	1.1298	1.1582	1.1596	1.1598
21/12/21	1.1272	1.1581	1.1596	1.1598
22/12/21	1.1324	1.1580	1.1596	1.1598
23/12/21	1.1320	1.1579	1.1596	1.1598
27/12/21	1.1329	1.1579	1.1596	1.1598
28/12/21	1.1314	1.1578	1.1596	1.1598

29/12/21	1.1337	1.1577	1.1596	1.1598
30/12/21	1.1318	1.1576	1.1596	1.1598

TABLE 10 – Predicted values of the LSTM, ARIMA, and DRW models in the 60-day horizon