Ensemble of temporal convolutional and long short-term memory neural networks apply to forecasting USDCOP exchange rate

Student: Juan Esteban Torres Marulanda
Advisor: Paula Maria Almonacid Hurtado

Maestría en ciencias de los datos y analítica
Universidad EAFIT

May 24, 2021

Abstract

This paper applies a neural network with ensemble of temporal convolutional network (TCN) and long short-term memory (LSTM) layers approach to forecast foreign exchange rates between the US dollar (USD) and Colombian Peso (COP) and obtain a better performance. This study provides evidence on the TCN and LSTM neural network model's effectiveness and efficiency in forecasting temporal series. It should contribute positively to developing theory, methodology, and practice of using an artificial neural network to develop a forecasting model for financial temporal series.

Keywords: LSTM, neural network, temporal convolutional, forecasting.

1 Introduction

Forecasting the USDCOP exchange rate has been one of the biggest challenges in Colombia (Rowland, 2014). Usually, traders apply analytical tools using past market data in an attempt to predict the rate's direction. Nevertheless, new techniques can help to improve the performance of this traditional methodology. In recent years, Deep Learning algorithms have improved predictions involving time series data (Makridakis, Spiliotis,

& Assimakopoulos, 2020). This field has also achieved remarkable results in other areas such image and speech recognition, natural language processing (NLP), autonomous vehicles, and games. Recent competitions like the M4 Competition, whose purpose is to learn how to improve forecasting accuracy in time series, have shown how hybrid models with traditional statistics and neural networks have obtained consistent results (Makridakis et al., 2020).

Therefore, the purpose and the contribution of this study is to develop a forecasting model applied to the USDCOP exchange rate in Colombia using a univariate time series with neural networks through ensemble temporal convolutional network with long short-term memory (LSTM). This paper has different sections as follows. After the introduction, Section 2 is a literature review of authors who speak about this. Section 3 gives a short review of the data. Section 4 introduces the method and describes it in a general sense. Section 5 evaluates the methodology, and finally, Section 6 discusses conclusions and future work.

2 Literature review

Since Meese and Rogoff (1983a, 1983b, 1988), it has been well known that exchange rates are challenging to predict using economic models (Rosi, 2013). Only a few monetary models with very long horizons have obtained good results. This is because movements of foreign exchange rates are generally non-stationary and quite random. The price input is not a desirable set, making forecasting difficult (Chen, Lou, & Chang, 2019). To predict the exchange rate, we can use linear or non-linear models, but the literature focus on linear models. Still, a few studies have focused on non-linear models (Diebold, X, & Nason, 1990), (Meese & Rose, 1991), (Chinn, 1991), (Mizrach, 1992), (Chinn & Meese, 1995); neural networks (Qi & Yangru, 2003); or (exponential) transition autoregressive models (ESTAR) (Rapach & Mark E, 2006). In recent decades, neural networks and other Machine Learning (ML) algorithms have been used successfully in forecasting different non-linear models. Their success has been with large representative datasets because ML algorithms can identify non-linear patterns and explore unstructured relationships without hypothesizing them a priori (Smyl, 2020). For this reason, neural networks are one of the most common approaches for forecasting financial time series (Chen et al., 2019). Some articles have shown the use of the neural network in predicting foreign exchange rates in different countries. For example, Panda and Narasimhan demonstrated the use of a neural network to make a one-step-ahead prediction of weekly Indian rupee/U.S. dollar exchange rate (Panda & Narasimmhan, 2007). (Jamal, 2005) applied a neural network model for representing the United States' exchange rate with two of its major trading partners, Canada and the Euro countries. (Dunis & Williams, 2002) examined and analyzed the use of neural network regression (NNR) models in foreign exchange (F.X.) forecasting and trading models.

Specially, the recurrent neural network has shown the ability to forecast EURUSD returns and added value as a forecasting and quantitative trading tool (Chen et al., 2019).

In particular, the *Long Short-Term Memory Networks* (LSTMs), composed of a *cell*, has an *input gate* that controls the extent to which a new value flows into the *cell*, a *output gate* that controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit, and a *forget gate* that controls the extent to which a value remains in the cell. This type of neural network is well-suited to classifying, processing and making predictions based on time series data. There can be lags of unknown duration between important events in a time series. Fisher and Krauss (Fisher & Krauss, 2017) demonstrated that an LSTM network could effectively extract meaningful information from noisy financial time series data. From 1967 until 1991, the Colombian peso was similar to the U.S. dollar at a pre-specified exchange rate and did not deviate significantly from this rate. In June 1991, the crawling peg regime was abolished following a sharp fall in international coffee prices and a deterioration in the trade balance. The central bank introduced an official crawling band regime in January 1994 to regain control over monetary variables after low real interest rates combined with substantial capital inflows. In September 1999, they dismantled the exchange rate band, and the exchange rate was allowed to float freely (Rowland, 2014).

Modeling and forecasting USDCOP exchange rates is a significant economic issue. Many entities have given outstanding attention to forecasting this rate because they help brokers and businesses make better decisions and provide valuable insight for government entities like the central bank responsible for overseeing its monetary system and preserving financial and payment systems stability. For this reason, Colombia has developed many different methodologies for predicting the exchange rate (Rowland, 2014).

According to the literature review, time series forecasting has improved through deep learning techniques (Smyl, 2020). This paper implements a USDCOP forecasting model ensemble convolutional and recurrent neural networks to predict a multi-step price range. First, we transform the data by individual time windows that collect the appropriate currency's historical information. Second, we ensemble a temporal convolutional layer and many LSTM layers that capture the currency's temporal history significant for predictions.

3 Data

The data include the representative exchange rate (TRM) of Colombia. It is based on the weighted average of the buy and sell foreign exchange rates for transactions completed on previous business. We downloaded the data from the Bank of the Republic of Colombia. We take business daily data, and for all missing observations, we use the previous day. The data set includes approximately 858 daily TRM from December 01, 2016 to March 16, 2020. The series of USDCOP is plotted in Figure 1. From the graph, the series appears as non-covariance stationery. Additionally, we plot log returns (Figure 7 in Appendix I) to understand the series better.

The currency has presented a substantial devaluation since the second quarter of 2018.

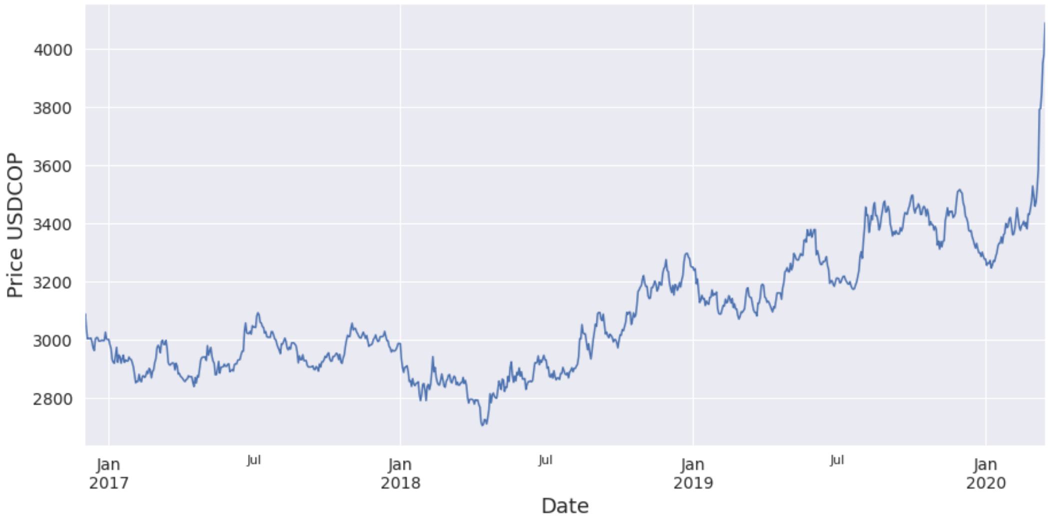


Figure 1: Price USDCOP

Colombia has what is known as a "free-floating economy" meaning that the value of its money is directly related to how much people are buying. This simple rule of supply and demand makes the USCOP exchange rate both interesting and important to track from a historical and predictive perspective. USDCOP rate fluctuations represent a risk to investors so it is important as a variable and we can try to predict using historical prices. In this paper we will try the forecasting exchange rate as univariate problem.

4 Methodology

This paper will forecast a trend in the foreign exchange rate for 3 days, which occurs when the exchange rate moves in a definable path over a specific time. According with our experience, we established a forecast target period, which is quite useful for traders for anticipating the currency trend in the short term. Traders found this useful because they made their investments over the long term. The objective of the developed LSTM neural network (Gers, Schmidhuber, & Cummins, 1999) is to predict the trend of the USDCOP exchange rate for up to 3 days ahead of the last data available. The output variable of the LSTM neural network is then the daily exchange rate, and the input data are collected each business day. The contribution of ensemble LSTM with a TCN layer seeks to improve the forecast.

In neural network applications, the input data are usually normalized or standardized into a range according to the neuron's activation function. In this case, normalizing is essential to improve the learning rate in the LSTM neural network (Smyl, 2020). In each input window across all epochs, we find a minimum and maximum USDCOP price to normalize the input values in a scale 0 to 1 (min-max normalization), and with this same information, we normalize the projection, which is our target. It is taken from the techniques literature review because the results improved grateful to data transformation, improving the learning rate.

The neural networks need three inputs: tensor dimension; window size, where we use 10 days after running cross-validation technique; and batch size with 16 windows. The output was established with target size of 3 daily prices. The hyper-parameters were chosen using some combination of reasoning, intuition, and back-testing as shown in Table 1. Cross validation was the main tool used for preventing over-fitting.

Table 1: Details of the architecture and tuning hyper-parameters used to ensemble neural network.

Hyper-parameter	**Tuning**	**Selection**
Hidden layers	2 - 3	3
TCN Activation function	relu - sigmoid	relu
Input slide window	5 - 10 - 20 days	10 days
Optimizer learning rate (log scale)	Scheduler $10^{-8} : 10^{2}$	10^{-1}
Optimizer momentum	0.5 - 0.6 - 0.8 - 0.9	0.9
Epochs	200 - 400 - 600	400
TCN 1D layer filters	8 - 16 - 32 - 40	16
TCN 1D layer kernel size	1 - 2 - 3	2
TCN 1D layer dilation rate	1 - 2 - 3	2
State size of LSTM layers	20 - 40 - 60 - 80	40

We made a neural network combination where first we used a temporal convolutional layer with one-dimension. This way, the kernel moves in one direction from the beginning of a time series towards its end. Then, we added the multiplication results together and applied a nonlinear activation function to the value. The resulting value becomes an element of a new *filtered* univariate time series, and then the kernel moves forward along the time series to produce the next value. Dilated operation is one of the benefits of using this temporal convolutional layer operation, and it also extract the correlation between the predicted variable and correlations that exist. All of this avoids overtraining (Smyl, 2020).

The temporal convolutional networks has two main constraints: the output of the network should have the same length as its input, and network can only use the information from past time steps. The main constraint is that when predicting the output for some time step, it can only use the inputs that have been observed previously. For its all convolution layers have the same length, with zero padding to ensure higher layers the same length as previous ones. Particularly, dilated convolutions it is expected that the networks can memorize long term information (He & Chao, 2019), which is we are looking forward in this paper. We shown in Figure 2 the convolution window with kernel size of 2.

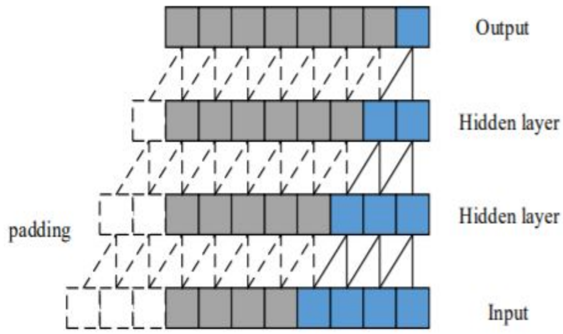


Figure 2: Convolutional Network- Visualization for: (He & Chao, 2019)

In the second step, we ensemble with two LSTM layers due to the advantage of the long-term memory in this kind of forecasting. The long-term memory usually calls the cell state because this allows for storage of information from previous intervals within the LSTM cell. The cell state is modified by the forget gate placed below the cell state and adjusted by the input modulation gate. The previous cell state forgets by multiplying with the forget gate and adds new information through the input gate's output. The forget gate is the remember vector. The save vector is usually called the input gate. Gates determine which information should enter the cell state/long-term memory. However, the significant parts are the activation functions for each gate. We shown in Figure 3 the LSTM function.

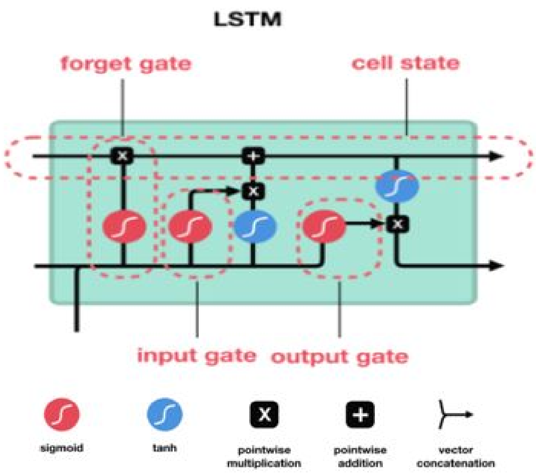


Figure 3: LSTM Cell - Visualization for: (Phi, 2018)

Finally, we ensemble the last LSTM layer with one dense layer possessing a number of neurons equal to the target size. To predict the USDCOP exchange rate with the recurrent neural network, we use the architecture that we explained before and show in Figure 4, where the optimization strategy was: data normalization with min-max normalization; optimization algorithm SGD with momentum; optimizer loss Huber and optimizer metric MAE. We have used five different layers combined into one ensemble network performing the final forecasting. One input layer, one hidden dilated temporal convolutional layer, an ensemble with two LSTM hidden layers and combines them into one final output layer.

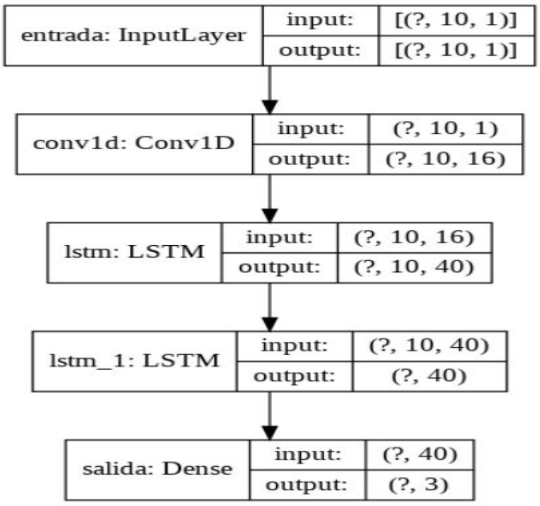


Figure 4: Architecture ensemble neural network

The metric used was mean absolute error (MAE) because, in contrast to conventional the mean square error (MSE) commonly adopted to approximate Gaussian errors for regression, MAE is more appropriate as a loss function for deep neural network-based vector-to-vector regression (Qi Jun, 2020) and is more robust to data with outliers which is common in USDCOP exchange rate. Additionally, mean absolute percentage error (MAPE) was used in the model selected to be contrasted, MAPE is commonly used as a loss function for regression problems and in model evaluation.

We use a stochastic gradient descent in our method for optimizing with a momentum of 0.9. We use 200 epochs for tuning the more sensible hyper-parameter, learning rate. However, in the learning where the most important hyper-parameter is challenging, a small value may result in a lengthy training process that could get stuck. For this reason, we used tuning with different learning rates (Table 1). In contrast, a value too large may result in learning a sub-optimal set of weights or a training process that is either too fast or unstable. For this reason, we used tuning with different learning rates (Figure 11 in Appendix II) and select the optimal rate across the callbacks option from the TensorFlow package.

We use the next plot (Figure 12 in Appendix II) to explore how the learning rate impacts the model's learning and learning dynamics rate. We show the loss function we want to minimize (Huber) and the metric mean absolute error (MAE) to judge the performance, and compare the behavior with different learning rates to select where it is most stable. The last step is to return to the original scale predictions made for each target size, which depend on the size of the windows used to predict.

5 Results

5.1 Data analysis

To better understand data and forecasting, we include descriptive statistics for the USD-COP (TRM) in Table 2. There are almost as many positive variations as negative ones, so the average change is almost zero in the returns. Prices have been oscillating between 2700 COP and 3500 COP with a normal distribution having mean 0% and volatility of 0.7% daily. The distribution and test of probability (Figure 8 in Appendix I) show us the distribution is not symmetric and is right-skewed, which is normal in financial time series.

Table 2: USDCOP exchange rate descriptive statistics.

Measure	**USDCOP close price**	**USDCOP Log return**
Mean	3085.82	0.03%
Standard deviation	211.47	0.70%
Minimum	2704.78	-2.60%
Q1 - 25%	2915.60	-0.38%
Q2 - 50%	3019.29	0.00%
Q3 - 75%	3240.01	0.42%
Maximum	4090.00	5.60%

Additionally, we plot auto-correlation and partial autocorrelation (Figure 9 in Appendix I) where we try to find the degree of similarity between a given time series and a lagged version of itself over successive time intervals. In both plots, we see auto-correlation for 23 days. Additionally, the decomposition (Figure 10 in Appendix I) involves thinking of a series as a combination of level, trend, seasonality, and noise components. Thus, it provides a useful abstract model for thinking about time series generally. We can see a remarkable tendency in this series for white noise and seasonal problems during time series analysis and forecasting. The main purpose of the analysis is to describe the data and find patterns that exist within it like a univariate analysis, not deal with causes or relationships with other variables like multivariate analysis.

5.2 Optimal model selection with cross-validation

In this section, we applied the cross-validation technique to select the best model. We use cross-validation to optimize the most sensitive hyper-parameters of the model. Specifically, the stochastic gradient descent (SGD), learning rate, and the size of the window's hyper-parameters were optimized as shown in Table 1. Concerning the state size of LSTMs or memory state, we work with a size of 40 because the size of the state was not a sensitive parameter, with values above 30 working well. The data set was separated into training and validation data, with the training data corresponding to the period between December 1, 2016 and February 29, 2020, while the validation data are between March 1, 2017 and March 15, 2020 and are exclusively for the final results where we use the optimal model selected in the process. The training data was split into 6 folds or time-based windows. Each fold, we divided range days into 3 years of training data with approximately 780 data points (business days) and 15 days of test data (with approximately 10 days being business days).

The windows that include the trainings and tests of the 6 folds are shown in Figure 5. We tune the learning rate (lr) hyper-parameter through the callbacks option of the TensorFlow package using 200 epochs, where the best neural networks were whit the lr of 10^{-1} in terms of lower loss.

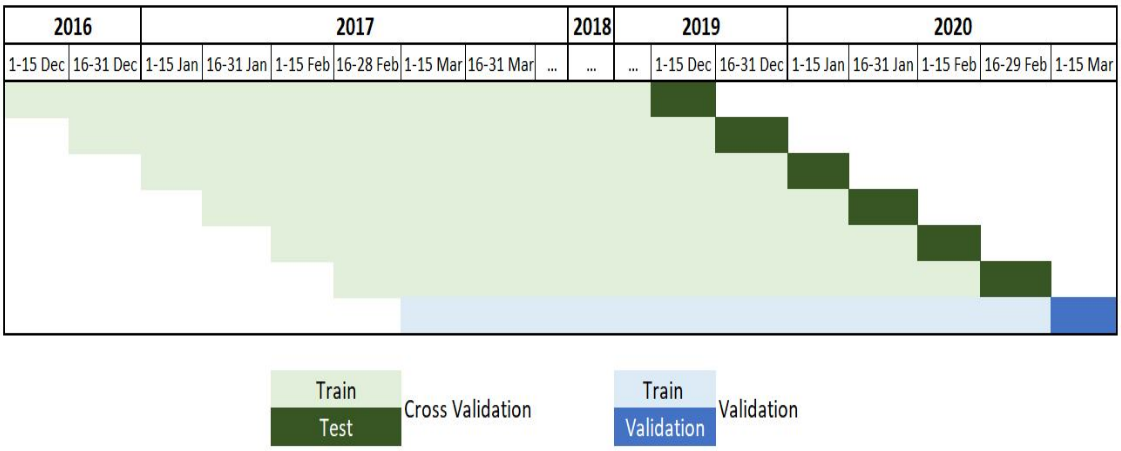


Figure 5: Train - Test

We test windows of 5, 10, and 20 days for the slide window hyper-parameter, and the steps of the windows entered into the network were configured with steps of 1 day. In terms of MAE the best results were with the neural network having a window size of 10 days where it won in 3 folds, as shown in Table 3.

In Table 4, we can see the best model results with a window size of 10 days obtained for each iteration and how 4 iterations of the 6 times evaluated won against a previous day's naive benchmark and LSTM neural network benchmark. Still, in Figure 13 and Figure 14 found in Appendix II, the model always seeks to predict a trend that can allow

Table 3: Cross validation result for ensemble neural network (TCN+LSTM) with slide windows 5, 10, 20 days, which is a sensitive hyper-parameter input to neural network. The evaluation metric was mean absolute error (MAE).

Models	**Ensemble NN Sw=5**	**Ensemble NN Sw=10**	**Ensemble NN Sw=20**
Slide Window - Date	**MAE**	**MAE**	**MAE**
Sw 1 - 1:15-dec-2019	245	216	189
Sw 2 - 16:31-dec-2019	196	335	307
Sw 3 - 1:15-jan-2020	242	93	268
Sw 4 - 16:30-jan-2020	544	433	252
Sw 5 - 1:15-feb-2020	349	315	500
Sw 6 - 16:29-feb-2020	247	207	367

the trader to operate and make better decisions. For example, in Figure 13, the MAE was the best; it follows several trends. It seeks to predict the trend very similar to what was shown in the real USDCOP series; it shows how it moves according to the series, while in Figure 14 the MAE was the worst, showing opposite trend in some predictions to what is expected. It is important to remember that in this iteration, we do not retrain the neural network with test data. Consistency is important in the context of finance, is the property of not having mutually contradictory evaluations at different points in time, this paper is looking forward for this consistency instead or minimizes the loss at some point.

Table 4: Forecasting cross validation results over test dataset between ensemble neural network (TCN+LSTM) and benchmarks models (Naive and LSTM neural network). Input slide window 10 days (best model ensemble NN over cross validation). We evaluate the final results with MAE and add MAPE metric to support consistency.

Models	**Ensemble NN**		**Benchmark naive**		**Benchmark LSTM**	
Slide Window - Date	**MAE**	**MAPE(%)**	**MAE**	**MAPE(%)**	**MAE**	**MAPE(%)**
Sw 1 - 1:15-dec-2019 fig:13	216	0.77	276	0.98	387	1.38
Sw 2 - 16:31-dec-2019 fig:14	335	1.00	168	0.50	502	1.48
Sw 3 - 1:15-jan-2020	93	0.32	126	0.43	145	0.49
Sw 4 - 16:30-jan-2020	433	1.18	243	0.66	340	0.91
Sw 5 - 1:15-feb-2020	315	1.16	320	1.18	392	1.45
Sw 6 - 16:29-feb-2020	207	0.76	211	0.77	277	1.02

This study's findings show that the neural network learns from past patterns, checking the assumption that ensemble TCN and LSTM layers have added value because it identifies the projection trend, which is very useful for traders. Additionally, in most tests, it shows an improvement compared to the benchmark naive. We are talking about a random walk where we know that the next time step will be a function of the previous time step, often called the naive forecast, or persistence model.

5.3 Results Validation Data

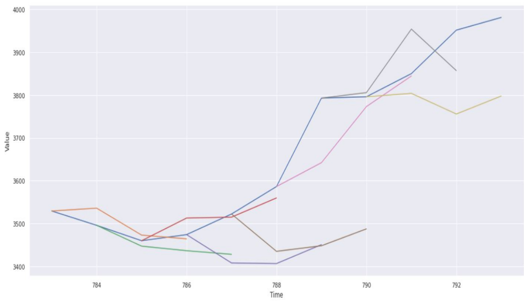
After running the cross-validation and selecting the best architecture, we implemented the neural network for the validation period between March 1 and 15 of 2020. We train the neural network with data between March 1 of 2017 and February 29 of 2020. Additionally, the training process used 400 epochs, which are when an entire data set is passed forward and backward through the neural network only once. We use the mean absolute error (MAE) to evaluate the neural network model and compare it with a naive model and LSTM NN model.

In the Table 5, through the MAE, the neural network model has better results of 845 compared to 942 for the naive model and 1006 for the LSTM NN model results. Additionally, in Figure 6, it is observed that when an upward trend began, the neural network captures this behavior better than the naive model, showing a similar behavior about the real trend that the trader would use to operate. It shows how the USDCOP exchange rate has changed over validation period (15 days) and each line forecast a trend in the foreign exchange rate for 3 days.

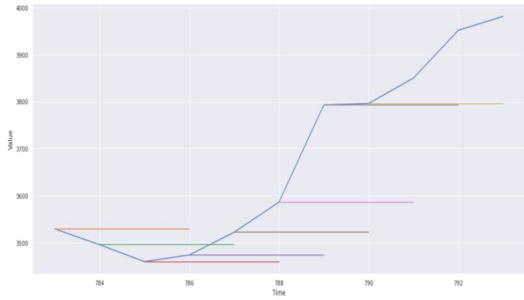
Table 5: Forecasting results over validation dataset between ensemble neural network (TCN+LSTM) and benchmarks models (Naive and LSTM neural network). Input slide window 10 days. We evaluate the final results with MAE and add MAPE metric to support consistency.

Models	**Ensemble NN**		**Benchmark naive**		**Benchmark LSTM**	
Slide Window - Date	MAE	MAPE(%)	MAE	MAPE(%)	MAE	MAPE(%)
Sw validation - 1:15-mar-2020	845	2.95	942	3.29	1006	3.47

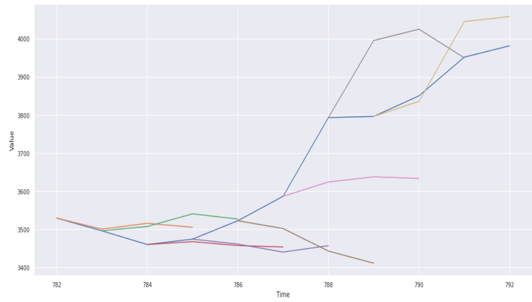
This paper shows a methodology applied toward a new complement tool to forecasting trends, which could be helpful to traders given the added value in being able to determine the exchange rate trend 3 days ahead to improve investment performance. It is valuable for speculation strategies and for hedging strategies. In the first one, speculators trade based on their guesses of where they believe the market is headed. For example, if a speculator believes that a stock is overpriced, they may short sell the stock and wait for the stock price to decline, at which point they will buy back the stock and receive a profit. At the same time, Hedgers reduce their risk by taking an opposite position in the market to what they are trying to hedge. The ideal situation in hedging would be to cause one effect to cancel out another.



(a) Ensemble NN



(b) Benchmark naive



(c) Benchmark LSTM

Figure 6: Validation 1 to 15 of march 2020

6 Conclusion and future work

- The forecasting of a random walk is still a complex task; it is pretentious to think that we could predict the future, but any improvement in the series trend helps the trader operate better in the market. It is worth analyzing other types of variables to add to the network with other techniques that isolate noise. One of the improvements could be to apply ensemble methods like stacking, which combine different models into one predictive model. In this case, the different models can be several recurrent neural networks with changed LSTM and temporal convolutional layers or other input variables. Next, each prediction from the models can be the inputs to a new predictive model that captures each model's advantages in order to upgrade the final prediction.
- The temporal convolutional layer and dilated LSTM alleviates the problem of learning long sequences. They manage to capture these short-term patterns and show the excellent potential of these methodologies in forecasting the trend. Adequate pre-processing of the data can improve performance and lead to better results.
- The process of finding the trend and seasonality of a time series is complex because the dollar has high volatility. Traditional models are looking for this. However, their models are rigid. Nevertheless, LSTM and temporal convolutional networks capture useful information from the past as well as non-linear behavior. In this case, the neural network's architecture is flexible to input data and can capture patterns from the long- and short-term without dependency from static trend or seasonally, which is sensitive to changes.
- Another possible upgrade for the model predicts a confidence interval for the USD-COP forecast inside the existing neural network architecture, which could be a useful tool for traders because they have become familiar with it when making decisions.

References

Chen, P. C., Lou, C. Y., & Chang, H. T. (2019). An empirical study of the artificial neural network for currency exchange rate time series prediction. *Department of Information Management, Yu-Da College of Business, Miaoli County, Taiwan 361*, 543-549.

Chinn, M. D. (1991). Some linear and nonlinear thoughts on exchange r. *journal of International Money and Finance*, *10*.

Chinn, M. D., & Meese, R. A. (1995). Banking on currency forecasts: How predictable is change in money? *Journal of international economics*, *38*.

Diebold, X, F., & Nason, J. A. (1990). Non-parametric exchange rate prediction. *Journal of International Economics*, *28*.

Dunis, C., & Williams, M. (2002). Modeling and trading the euro/us dollar exchange rate: Do neural network models perform better? *Derivatives Use, Trading and Regulation*, *8*.

Fisher, T., & Krauss, C. (2017). Deep learning with long short-term memory networks for financial market predictions. *FAU Discussion Papers in Economics*.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with lstm. *Neural Computation*, *12*, 2451–2471.

He, Y., & Chao, J. (2019). Temporal convolutional networks for anomaly detection in time series. *Journal of Physics: Conference Series*, 1-2.

Jamal, A. (2005). The neural network and exchange rate modeling. *International Journal of Management*, *22*.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*, 54-74.

Meese, R. A., & Rose, A. K. (1991). An empirical assessment of non linearities in models of exchange rate determination. *Review of economic Studies*, *58*.

Mizrach, B. (1992). Multivariate nearest-neighbor forecasts of ems exchange rates. *Journal of aplied econometrics*, *7*.

Panda, C., & Narasimmhan, W. (2007). The forecasting exchange rate better with artificial neural network. *Journal of Policy Modeling*, *29*.

Phi, M. (2018). *Illustrated guide to lstm's and gru's: A step by step explanation.* https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

Qi, M., & Yangru, W. (2003). Nonlinear prediction of exchange rates with monetary fundamentals. *Journal of empirical finance*, *10*.

Qi Jun, S. M. M. X., Du Jun. (2020). On mean absolute error for deep neural network based vector-to-vector regression. *IEEE SIGNAL PROCESSING LETTERS*.

Rapach, D. E., & Mark E, W. (2006). The out of sample forecasting performance of non-linear models of real exchange rate behavior. *International Journal of forecasting*, *22*.

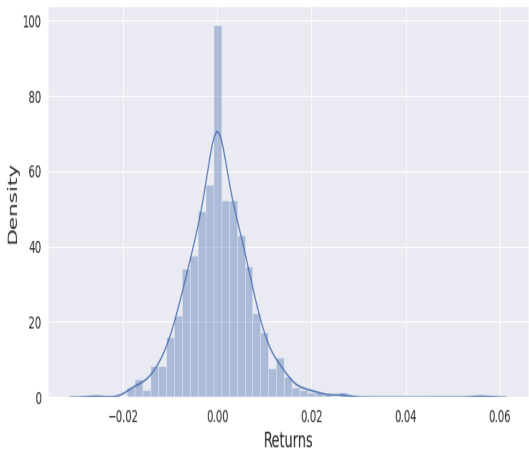
Rosi, B. (2013). Exchange rate predictability. *Journal of Economic Literature*, *51*, 1063-1119.

Rowland, P. (2014). Forecasting the usd/cop exchange rate: A random walk with a variable drift. *Banco de la Republica*.

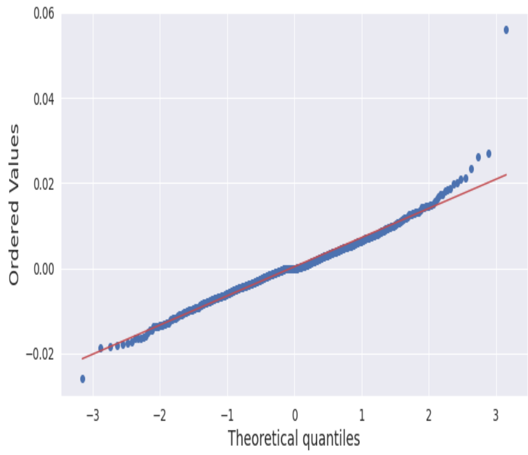
Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *30*, 75-85.

Appendix I: Statistics data analysis USDCOP

Figure 7: Daily log returns USDCOP

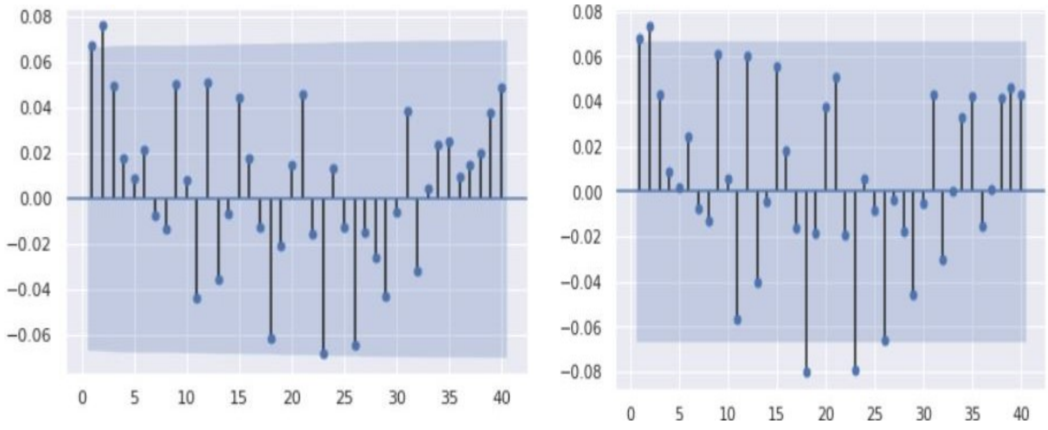


(a) Returns Histogram



(b) Probability plot

Figure 8: Returns distribution



(a) ACF (b) PACF

Figure 9: Auto-correlation and Partial Auto-correlation

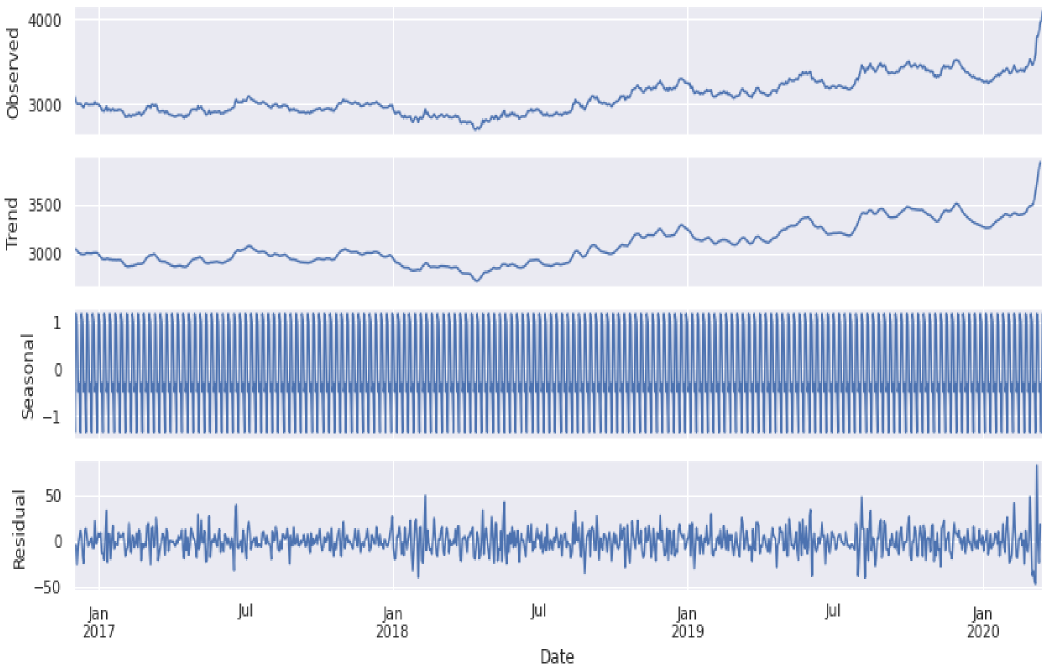


Figure 10: Decomposition USDCOP Time Series

Appendix II: Evaluation model

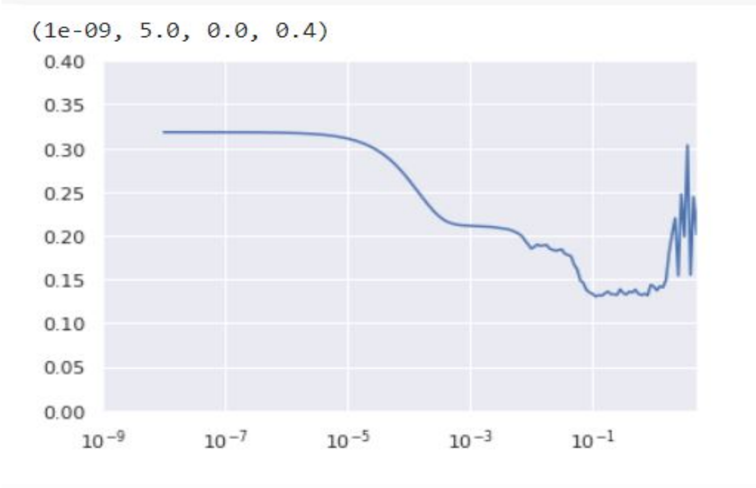


Figure 11: Learning Rate vs Loss

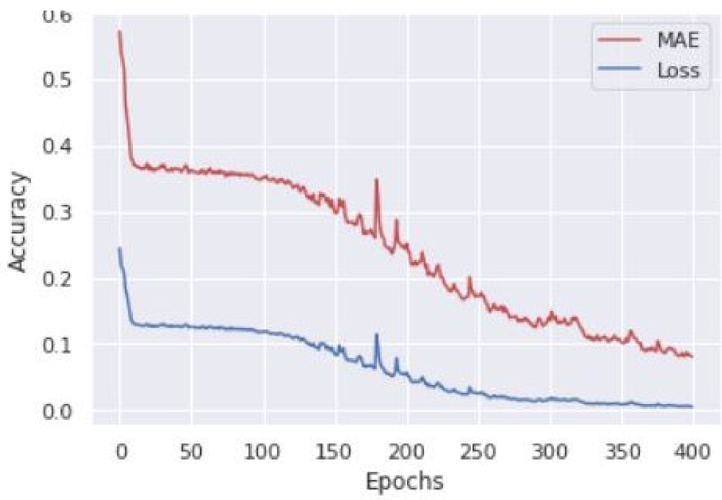
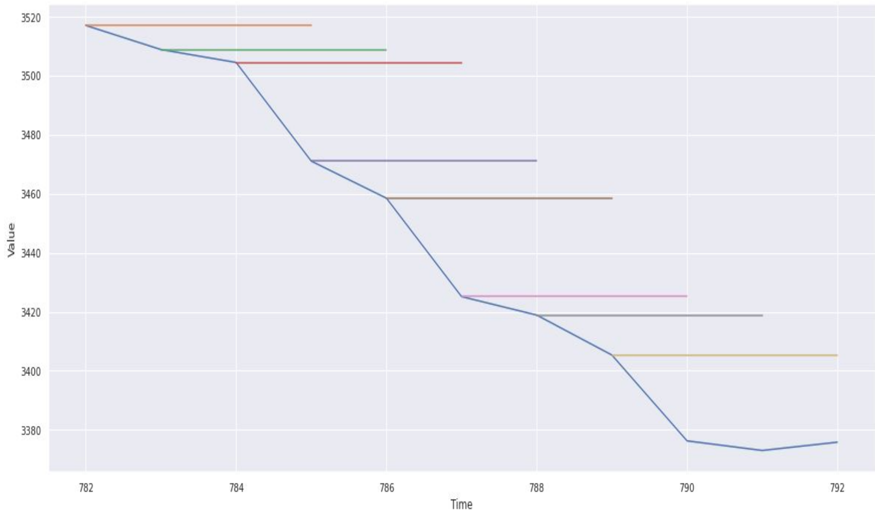
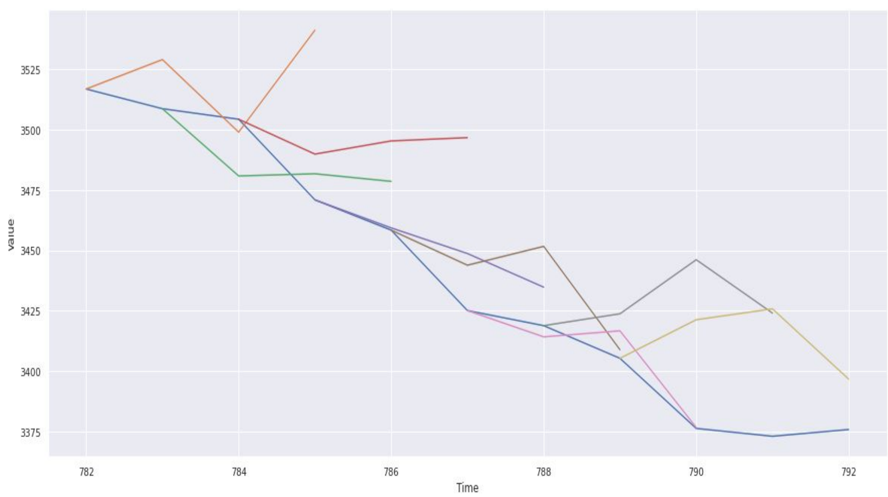


Figure 12: Epoch vs Loss/MAE

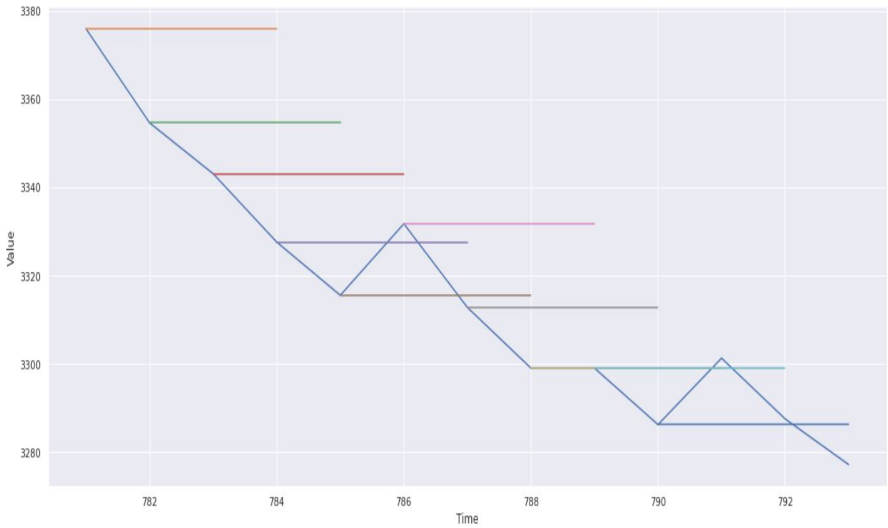


(a) Benchmark naive

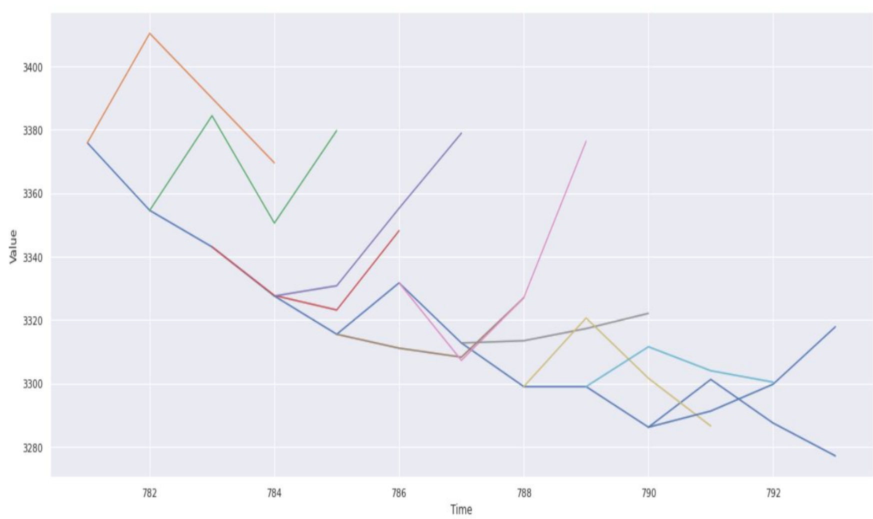


(b) Ensemble NN Model

Figure 13: 1 to 15 of December 2019



(a) Benchmark naive



(b) Ensemble NN Model

Figure 14: 16 to 31 of December 2019