CAPSTONE 2 MILESTONE REPORT

Problem Statement

Foreign exchange rate forecasting is essential for managing foreign exchange risk. Financial institutions and businesses with exposure to foreign currency transactions, constantly seek to manage this risk. Foreign exchange forecasting is also essential for currency traders both at an institutional level and at the individual level.

Predicting foreign exchange rate has been a challenging task for traders and practitioners in financial markets, and this project attempts to examine the effectiveness and performance of ARIMA (SARIMAX) and Recurrent Neural Networks (LSTM) in predicting Foreign Exchange rate.

Time series forecasting models support the assumption that past patterns in data can be used to forecast the future. The dataset used in this project is the EURUSD foreign exchange rates from 2000 to 2019 dataset, and the project attempts to predict EUR/USD exchange rate for the next 10 years, using SARIMAX Time series forecasting method.

LSTM

LSTM is a special kind of RNN composed of a set of cells with features to memorize the sequence of data. The cell captures and stores the data streams. Further the cells inter-connect one module of past to another module of present one to convey information from several past time instants to the present one. Due to the use of gates in each cell, data in each cell can be disposed, filtered, or added for the next cells.
Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. This resolves the vanishing gradient problem of RNN. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using backpropagation. In an LSTM network, three gates are present:

1. Input gate: discover which value from input should be used to modify the memory.
2. Forget gate — discover what details to be discarded from the block.
3. Output gate — the input and the memory of the block is used to decide the output.

I will be using the multi-layered LSTM recurrent neural network to predict the last couple of years of exchange rate

SARIMAX Model

The Autoregressive Integrated Moving Average is one of the most widely used forecasting methods for univariate analysis, but it does not support time series with seasonal component. As a result of this, the SARIMAX extension of the ARIMA model was used in this project, as it explicitly models the seasonal element in univariate data. ARIMA models work on the following assumptions :

The data series is stationary, which means that the mean and variance should not vary with time. A series can be made stationary by using log transformation or differencing the series.

The data provided as input must be a univariate series, since ARIMA uses the past values to predict the future values.

ARIMA consists of three trend elements that require configuration:

- ARIMA supports the autoregressive element of time series data i.e. the number of lag observations. It is defined by the autocorrelation function (ACF) and denoted by ‘p’
- ARIMA also supports Integrated element of time series data i.e. the degree of differencing and is defined by the stationary test of a time-series. We will be using the Dickey-fuller test for this. It is denoted by ‘d’ in the ARIMA model.
- Lastly ARIMA supports the Moving average element of time series data and is defined by the partial correlation function (PACF). It is denoted by ‘q’ in the ARIMA model.

SARIMA extends the seasonality component of Timeseries data (that is a time series with repeating cycle). It adds four other hyperparameters to the Arima model in order to deal with the Seasonal component of the time series as shown below:

P: Seasonal autoregressive order.
D: Seasonal difference order.
Q: Seasonal moving average order.
m: The number of time steps for a single seasonal period such as daily, weekly etc.

Together, the notation for an SARIMA model is specified as: SARIMA(p,d,q)(P,D,Q)m.

EXPLORATORY DATA ANALYSIS

Dataset

Consists of daily opening exchange rates of eurusd downloaded from barchart.com, from year 2000 to year 2019.

Data Cleaning

Data supplied from website is clean and complete and in the required format. I carried out the following checks to determine data quality:

I examined completeness and data types. I also examined the summary statistics of the dataset. All checks revealed that the dataset is clean and is in the required format to be used in my models.

Statistical Analysis

I tested two hypotheses as part of the exploratory data analysis, to determine the stationarity of data and also to determine whether the data set is Gaussian or not.

Stationarity:

For a timeseries to be considered stationary, it is expected that the mean and standard deviation should be constant which means the data should not have trend and seasonality.

I used the Dickey–Fuller test to test the null hypothesis that a unit root is present in an autoregressive model. A stationary series has constant mean and variance over time. Rolling average and the rolling standard deviation of time series do not change over time.

Dickey-Fuller test Null Hypothesis (H0): Time series has a unit root, meaning it is non-stationary.
Alternate Hypothesis (H1): Suggests the time series does not have a unit root, meaning it is stationary.

Hypothesis 1:
p-value > 0.05: Accept the null hypothesis (H0), the data has a unit root and is non-stationary.¶

p-value <= 0.05: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

Result of hypothesis 1:
From the above results, the Standard deviation is stationary, while the mean is not. We will reject the null hypothesis H0, the data does not have a unit root and is stationary.

The results above can also be observed in the charts below:

Chart 1: Shows time series before decomposition:

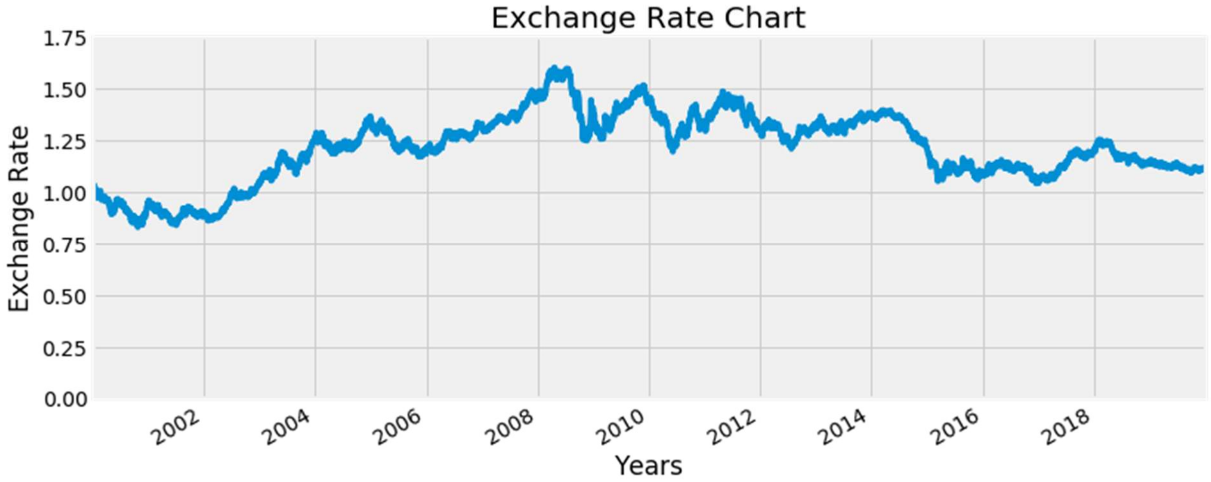


Chart 2: Shows time series after decomposition

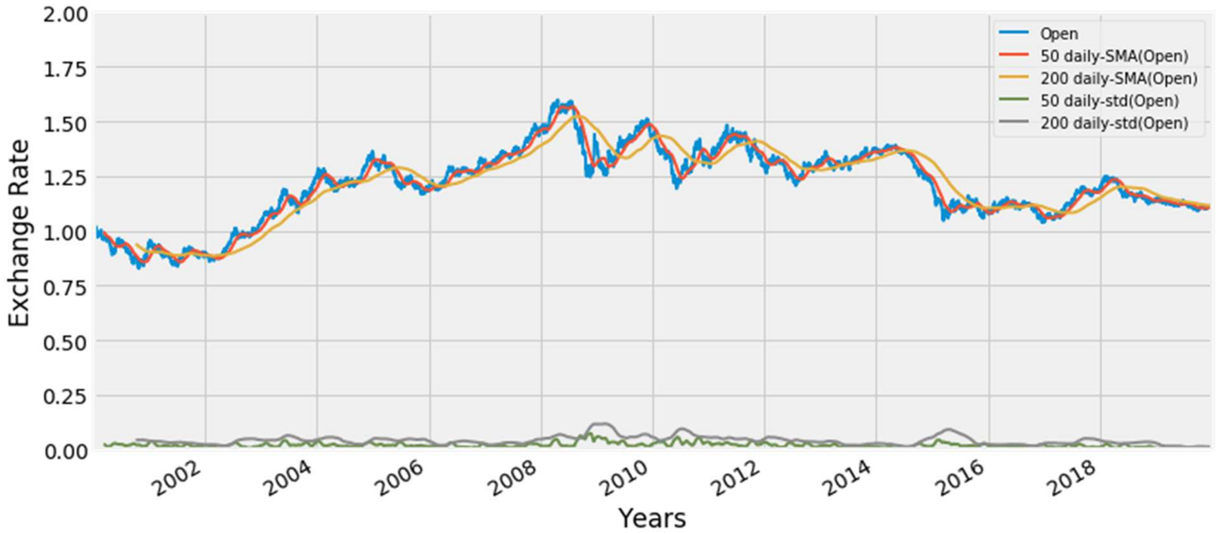
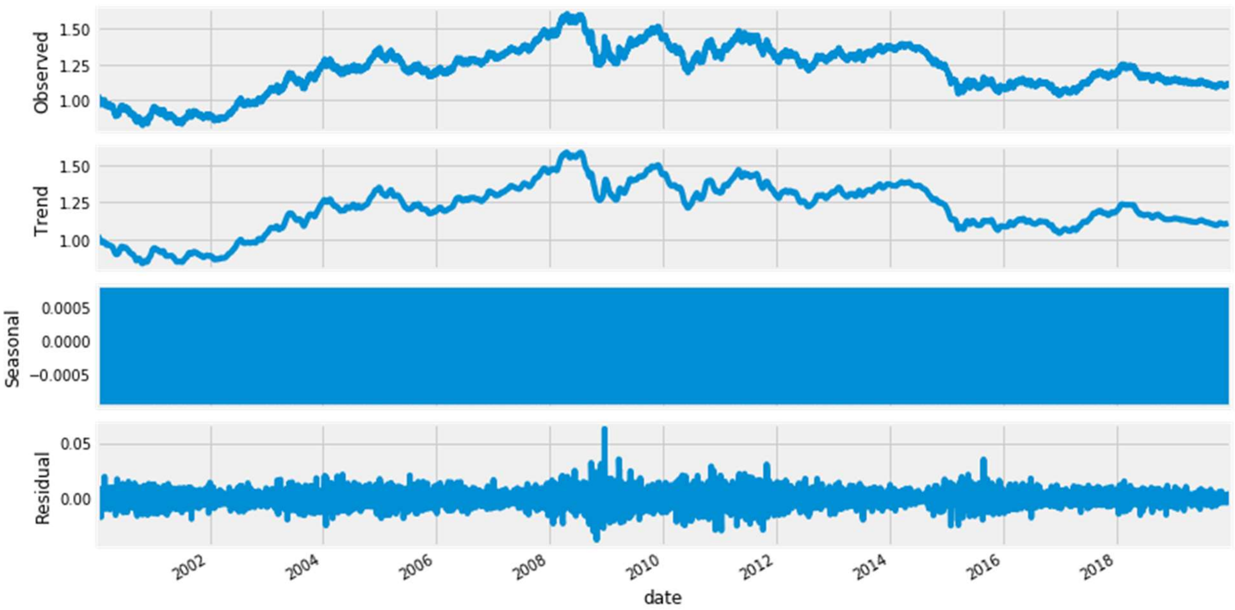
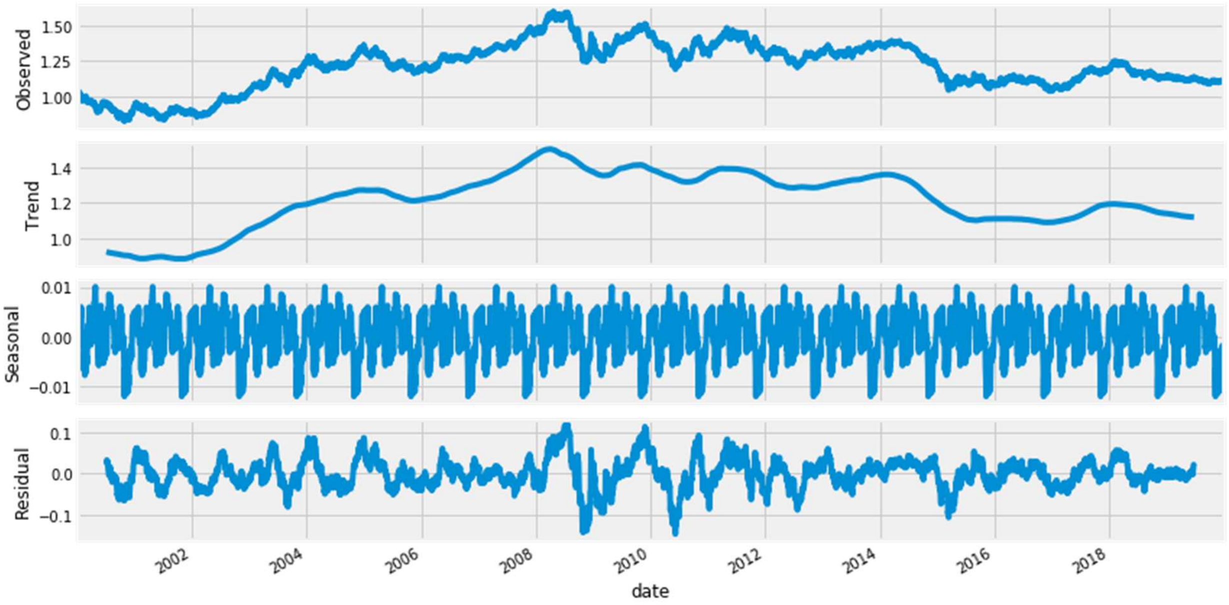


Chart 3 below shows the different components of the time series namely, the observed time series, the trend element, the seasonal element, and the residual element. For the frequency of seasonality, I used a weekly frequency of 52 and monthly frequency of 12 to show the impact on seasonality.

For the project I used a monthly frequency of 12, and m=12 because of system capacity issues. The model could probably have been improved by examining different frequencies.





From the chart above, the standard deviation is constant, however the mean trends in the same direction as the original time series.

Statistical Normality Test

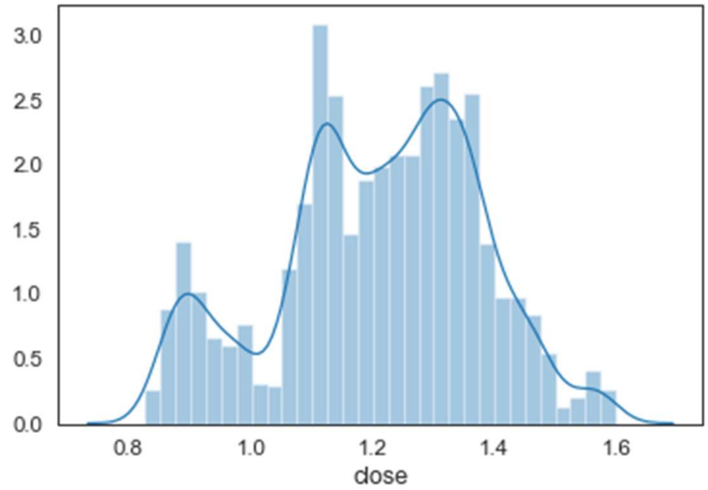
This test quantifies whether data was drawn from a Gaussian distribution, using the D’Agostino’s K² Test.

The Null hypothesis (H0) suggests that the data is gaussian and is therefore normal
The Alternate Hypothesis (H1), suggests that the data is not gaussian and is therefore not normal.

Hypothesis 2
For alpha = 0.05
p <= alpha: reject the null hypothesis (H0) , Data does not look Gaussian, therefore is not normal.¶
p > alpha: fail to reject null hypothesis (H0), Data looks Gaussian and normal

Result of hypothesis 2:
The test result reported a p=value of 0.000, which is lower than 0.05. From the statistics the data does not look Gaussian, we will therefore reject the null hypothesis H0.

Kurtosis test
Another test that determines the normality of the datasets is the kurtosis test. The kurtosis of the distribution below is less than zero and is light tailed. The distribution is fairly symmetrical and normal.



Statistics=137.136, p=0.000
Kurtosis of normal distribution: -0.43668157335097213
Skewness of normal distribution: -0.28722463936355497

It is important to ensure that the residuals of the model are uncorrelated and normally distributed with zero-mean. Where the seasonal ARIMA model does not satisfy these properties, it is a good indication that it can be further improved.

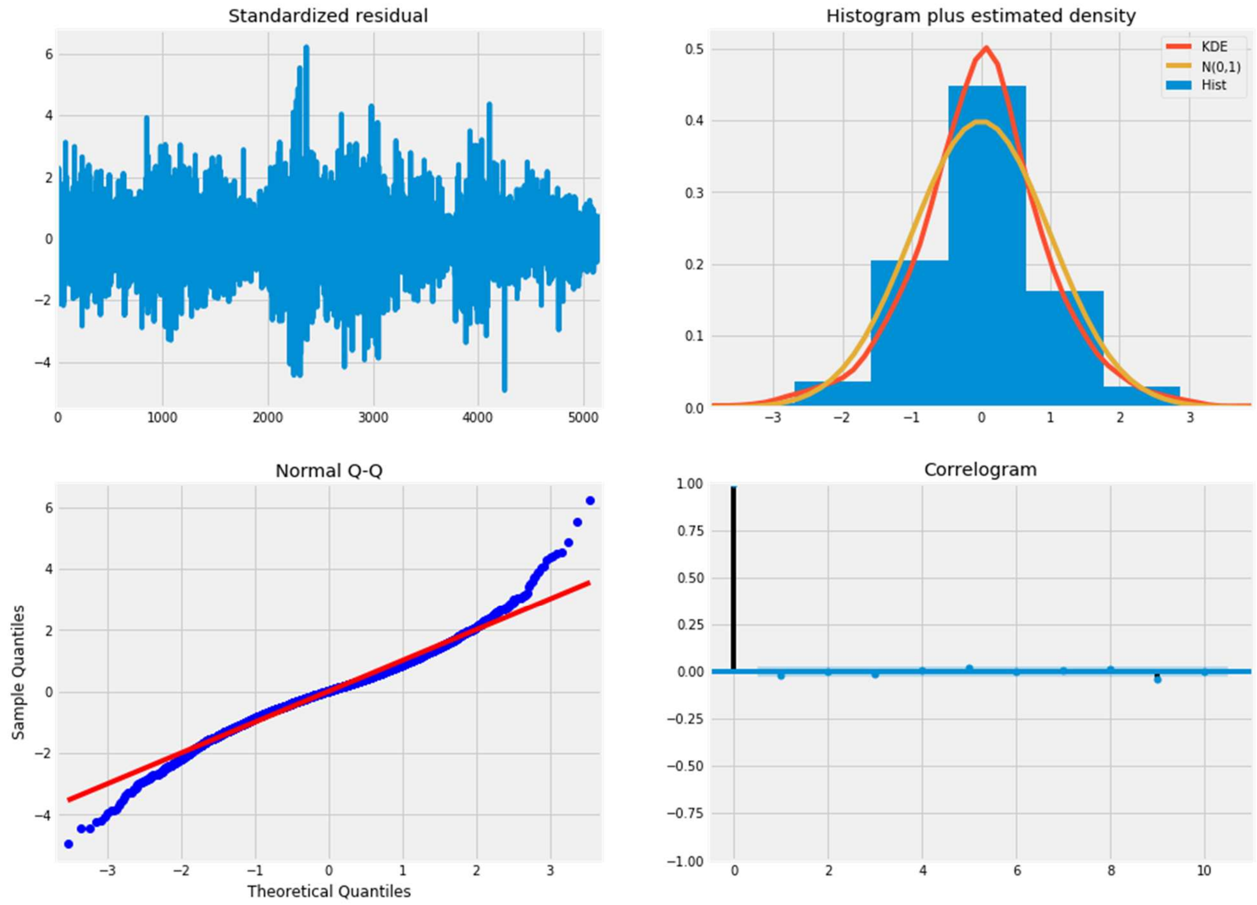
In this case, our model diagnostics below suggests that the model residuals are normally distributed based on the following:

In the top right plot below, the red KDE line follows closely with the N(0,1) line (where N(0,1)) is the standard notation for a normal distribution with mean 0 and standard deviation of 1). This is a good indication that the residuals are normally distributed.

The qq-plot on the bottom left shows that the ordered distribution of residuals (blue dots) follows the linear trend of the samples taken from a standard normal distribution with N(0, 1). Again, this is a strong indication that the residuals are normally distributed.

The residuals over time (top left plot) don’t display any obvious seasonality and appear to be white noise. This is confirmed by the autocorrelation (i.e. correlogram) plot on the bottom right, which shows that the time series residuals have low correlation with lagged versions of itself.

Those observations lead us to conclude that our model produces a satisfactory fit that could help us understand our time series data and forecast future values.



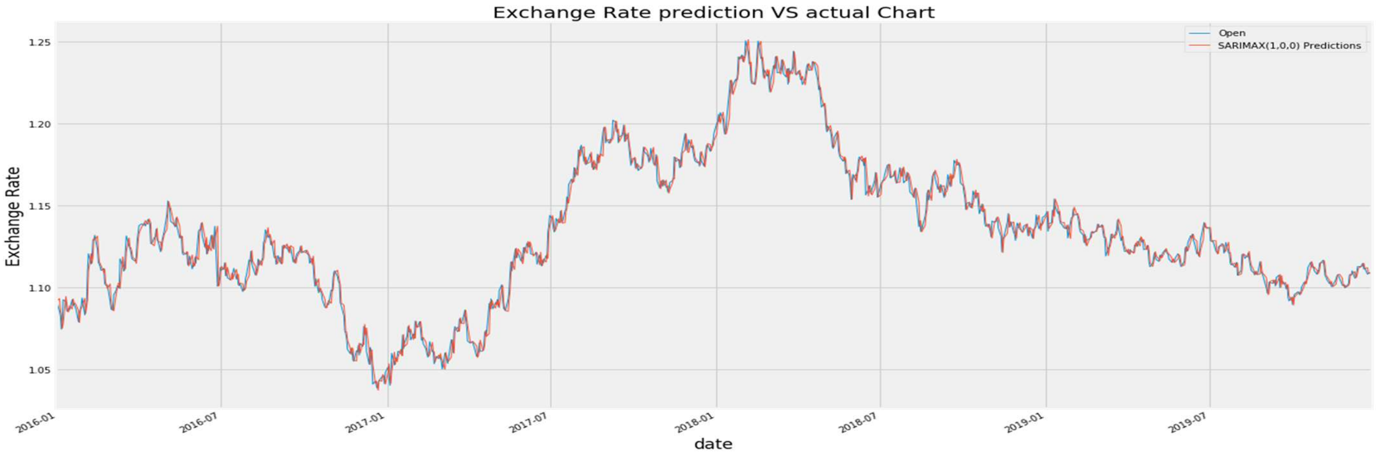
MACHINE LEARNING: SARIMAX

The dataset was split into training and test set using 80% split in favor of training set, where the training set is the first 80% of data, while the test set is the last 20% of data.

The auto ARIMA function carried out a stepwise search, to generate the optimal parameter values for the Arima Model, and results of the test is as follows: Model: SARIMAX(1, 0, 0)x(2, 1, 1, 12)

Fit Auto ARIMA: The model was fitted on univariate series as only the “Open” Column of the dataset was considered. Other columns of the dataset were dropped.

Predict values on validation set: Make predictions on the validation set by plotting predictions against known values as shown below:



Calculate Root Mean Square Error (RSME): This checks the performance of the model using the predicted values against the actual values: The RMSE of this model is: 0.005906521982

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are, and is measured by:

$$RMSE = \sqrt{(f - o)^2}$$

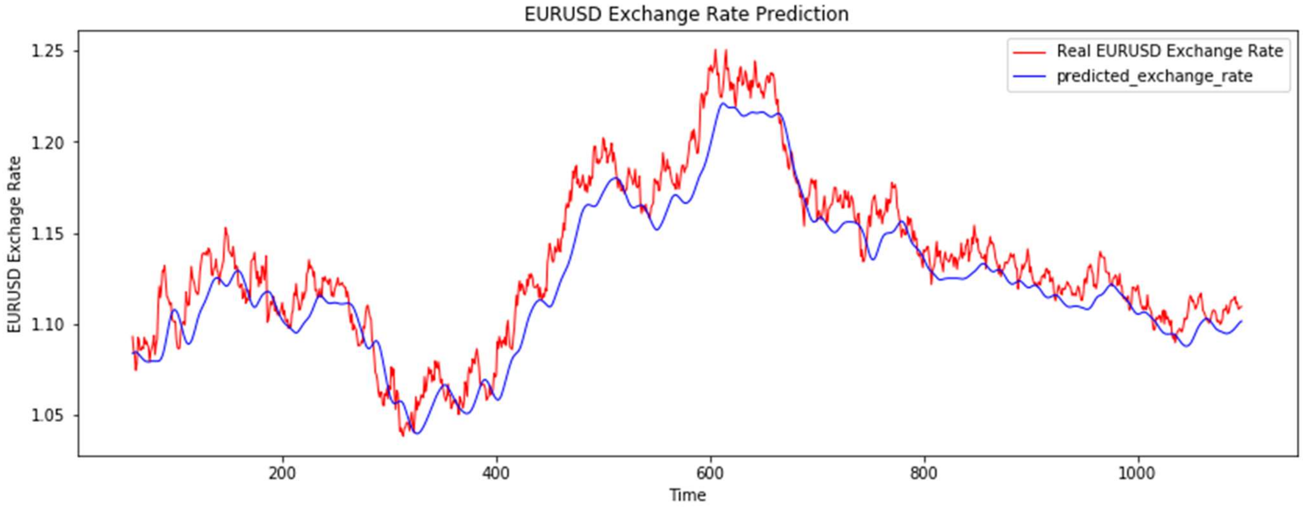
Where:
f = forecasts (expected values or unknown results),
o = observed values (known results).

LSTM RECURRENT NEURAL NETWORKS

The dataset was split into training and test set using 80% split in favor of training set, where the training set is the first 80% of data, while the test set is the last 20% of data.

The model was fitted on univariate series as only the “Open” Column of the dataset was considered. Other columns of the dataset were dropped.

Predict values on validation set: Make predictions on the validation set by plotting predictions against known values as shown below:



Calculate Root Mean Square Error (RSME): This checks the performance of the model using the predicted values against the actual values: The RMSE of this model is: 0.01606629

Conclusion

Comparing the root mean square error for the ARIMA model, and the LSTM Model, the ARIMA model had a lower RMSE of 0.005906521982 as compared with 0.01606629 generated by the LSTM Model which suggests in this instance that the ARIMA model performed better in terms of minimizing errors between the test set and the predictions.

Possible areas of improvement / optimization of the ARIMA and LSTM models include:

The ARIMA model could be further improved by experimenting with various frequencies and ‘m’ values in order to determine the optimal values that would optimize the model. I was not able to do this because of system capacity issues.

The LSTM Model could be further improved by experimenting to obtain the optimal batch size / epoch and patience values. A model loss chart could be added to determine optimal levels.

These two models could be useful for determining the turning points in the market. It is not advisable to use specific prediction for investment purposes; however, they can be used as a guide, along with other technical and fundamental analysis to develop a point of view of the EURUSD Exchange rate.

Source:
www.barchart.com
https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/
https://www.analyticsvidhya.com/blog/2018/08/auto-arima-time-series-modeling-python-r/
https://www.statisticshowto.datasciencecentral.com/rmse/
https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e