NORTH SOUTH UNIVERSITY

Group Members:

Mohammed Adib Khan (1430420042)

Sarwat Islam Dipanzan (1510117042)

Rifat Arefin Badhon (1511738042)

Niyaz Bin Hashem (1510120042)

Faculty:

Syed Athar Bin Amir (SAA3)

Course:

CSE498R - Directed Research

15th May 2019

Finding Profitable Entries In The Forex Market Using Machine Learning

Abstract — Predicting the trend or price action of a Foreign Exchange (Forex) instrument/currency pair is a challenging task. Analyzing the Forex market data, it is very difficult to build a model which gives an accurate prediction. In this study we have developed three models using Convolution Neural Network (CNN), Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA) and analyzed their performance to predict the opening price of 1 hour candlesticks for EURUSD pair (the most highly traded pair in the market) and its market trend direction. We tried to approach the problem through two different perspectives - a classification between when to take an entry and when to not react to the market and to predict the opening price of future candlesticks itself. Classification results were not in the acceptable range because when compared to taking actions versus not reacting to the market, the latter outnumbered the former by almost 6.7 folds resulting in a data imbalance for the model to train on properly. The regression approach worked very well with LSTM for predicting the opening price of future candlesticks, but when the same data was fed into ARIMA, it failed to predict the opening prices correctly because ARIMA puts more importance on the long term (old) trends than the present or latest trends market trends.

TABLE OF CONTENTS:

INTRODUCTION

Predicting Forex instrument price is considered a challenging task because Forex markets generally display a non-linear and nonparametric system. This system can also be regarded as noisy and deterministically chaotic, in other words, there is no certain pattern to predicting if the system will have a positive or negative swing in the context of a live market. The Forex market is inherently deterministic chaotic because there is “randomness” as associated with chaotic complex systems. This can be compared to other such systems such as weather patterns, economic markets, and biological systems where predicting outcomes can have arbitrary results. The Forex market is an important sector for many countries, especially for those who import, export or do both in massive volumes such as USA, China, Japan, etc. Forex market rate plays a huge factor in international trades because this is also a part of a business's cost. With a high volume of trades, even a slight increase or decrease of a currency pair value could mean a significant amount of money. This especially means a huge deal to big financial institutes such as hedge funds, banks, etc. Trend forecasting has become an essential topic for Forex traders, investors and the authority that are related to the Forex market business. The trend of a market depends on many things such as liquidity, various economic indices, human behavior, news related to the stock market, etc. All this together controls the behavior of trends in a Forex market. The behavior of trend can be analyzed by using technical tools, parametric pricing methods or a combination of these procedures.

Many machine learning techniques have been used in recent times such as Support Vector Machine (SVM), Multilayer Perceptron (MLP), Recurrent Convolutional Neural Network (RNN), etc. to predict stock market instrument prices. Neural networks and Long Short-Term memory are among the most feasible algorithms for predicting such behavior due to an innate advantage in time-series prediction. At present deep learning is making quite an impact in the stock market sector. But, barely any work has been done for the Forex market by utilizing machine learning approaches.

There are a lot of currency pairs or instruments to choose from, but most of the pairs which have lower liquidity are prone to high manipulation. Spread (price gap between ask and bid) also plays an important role in Forex trade because it introduces extra costs on both sides. Due to such reasons the most highly traded pair, EURUSD, was selected for this research first because it's spread is the lowest in the market, secondly manipulation in price for this pair is going to be the most costly and not worth the cost for any manipulator in monetary terms compared to other pairs due to huge trade volume, and thirdly because noise in this pair should be comparatively lower than others due to its low volatility. Our choice of time frame was 1-hour candlesticks because any lower than 1 hour would mean a lot of noise, which gets averaged out on a larger time frame such as 1 hour, whereas any higher than 1 hour would cause important trend movements to go undetected. This also makes the dataset noticeably small to work with.

RELATED WORK

Forex Market prediction is a challenging task and very few work has been done in this sector with machine learning techniques. The stock market, on the other hand, has seen a lot of attention from researchers using various techniques and algorithms. Studying related literature provided an understanding of the underlying concept of LONG (taking a market position at the buy side) in Forex market since it works exactly the same way as a stock market's buy and sell strategy (buy at a low price and sell at a high price to make a profit). The use of machine learning in this field is daunting as the prediction tends to fade due to various factors influencing the market. Artificial Neural Networks (ANN) mainly used as they tend to show promising results [1]. With a hybrid approach of ANN and Rough Set (RS), they were able to pull off an astonishing accuracy of 97% on Dhaka Stock Exchange price action trend prediction. A study on stock price using ANN, SVM and Random Forest (RF) showed that the prediction accuracy increases significantly with RF if the input data are represented as continuous values (such as that we experience in a real-time market), whereas it increases overall if the pricing inputs are represented as trend deterministic data (i.e. not training on direct price action data) [2]. Recurrent Convolutional Neural Network (RNN) was used to detect stock price by Bo Xu where they have interestingly used a layer of Convolutional Neural Network (CNN) to analyze market sentiments through yahoo news and predict the price action by using a Long Short-term Memory (LSTM) layer [3]. A comparison between SVM, CNN, and their RNN model was carried out and they found that their RNN model outperforms the other two's accuracy by 8% and 3% respectively. Deep learning is a new popular concept in the realm of machine learning, Huy D. Huynh used deep neural networks to predict the stock price to produce better results [4]. They have

experimented with RNN, LSTM, Gated Recurrent Unit (GRU) and introduced the Bidirectional Gated Recurrent Unit (BGRU). BGRU was also used for training "words" which were financial news encoded as real-valued vectors. Their findings were that RNN, LSTM, and GRU all performed pretty much the same (accuracy- 55%, 58%, 58% respectively). But, BGRU outperformed them all with 59% accuracy when tests were done on a generalized set of stocks. But it showed massive improvements when it was tested with individual stocks, e.g. Wal-Mart, accuracy jumped up to 66% which was at least 3.5-4% higher than the other models with the same tests. The amount of work done in this field is very diverse and we plan to use two key algorithms to predict price and entry points using Dukascopy's Forex instrument data [5].

DESCRIPTION OF DATA & PREPROCESSING

Forex market data was collected through Dukascopy's open source online portal (https://www.dukascopy.com/plugins/fxMarketWatch/?historical_data) [5]. Dukascopy is a Swiss Banking group and a world-renowned Forex broker who has pioneered the Forex trading platform for both institutions as well as retail customers [6]. Our data collection period extends from 1st March 2018 to 1st March 2019 (a total of 6249 rows - meaning 6249 hours of active trading data; excluding holidays' data). Each record consist of 6 columns - candlestick timestamp, opening price, high price, low price, closing price, and trade volume. Sample data is provided in figure 1. The dataset consists of 1 hour of candlesticks time frame data.

Gmt time	Open	High	Low	Close	Volume
01.02.2018 00:00:00.000	1.24173	1.24186	1.2417	1.24176	145.02
01.02.2018 00:01:00.000	1.24176	1.24191	1.24176	1.24186	156.33
01.02.2018 00:02:00.000	1.24186	1.24218	1.24186	1.24197	172.4
01.02.2018 00:03:00.000	1.24197	1.24216	1.24194	1.24197	224.07

Figure 1. EURUSD 1hour timeframe candlesticks raw data

People express their appreciation for an instrument in the market through price. Price of an instrument is comparable to the voting result of an election. We derived the market trend which is comparable to an election poll.

Over the past years, mathematicians and economists have developed hundreds of formulae to help them assist with predicting trends of markets. These set of formulae or tools is known as technical analysis (TA) indicators. A lot of such indicators have become obsolete and fails to work properly in the present market, but numerous indicators have been developed to tackle such changes in the present scenario. TA indicators are usually classified into two categories-

oscillators (used to determine if an instrument is overbought or oversold) and moving averages (used for determining and comparing long term vs short term trends). Our datasets were divided into three categories:

1. Only oscillator indicators.
2. Combination of moving averages and oscillators indicators.
3. Raw price action candlesticks data input.

We decided to remove some of the indicators because they do not give suitable outcomes in all scenarios, some of them are good for long term predictions whereas others are far better at short term predictions. A few of those indicators work better for an emerging market and consequently, some fail in that same sector. These are the reasons for choosing a combination of indicators of different categories so that a better accuracy could be achieved. The indicators used along with their interpretations have been described below.

1) *Simple Moving Average (SMA)*

SMA is an arithmetic moving average calculated by adding recent closing prices and then dividing that by the number of time periods in the calculation average. SMA is useful for working with long term market trends [7]. We have used a slow SMA of period 200 and a fast SMA of period 50 (this setting provides an excellent trend comparison [7]). Our main target out of it was to provide a trend comparison for long term and short term basis. Numerous

$$\text{SMA} = \frac{A_1 + A_2 + ... + A_n}{n}$$

where:

A_n = the price of an asset at period n

n = the number of total periods

2) *Exponential Moving Average (EMA)*

EMA is a type of moving average that places a greater weight and significance on the most recent data points. EMA react more sensitive to recent price changes than an SMA, which applies an equal weight to all observations in the period. EMA is useful for working with short to midterm market trends [8]. We have used a slow EMA of period 26 and a fast SMA of period 12 (this setting provides an excellent trend comparison [8]). Our main target out of it was to provide a trend comparison for midterm and short term basis by weighing down more towards recent prices rather than old prices.

$$EMA_{\text{Today}} = \left(\text{Value}_{\text{Today}} * \left(\frac{\text{Smoothing}}{1 + \text{Days}}\right)\right) + EMA_{\text{Yesterday}} * \left(1 - \left(\frac{\text{Smoothing}}{1 + \text{Days}}\right)\right)$$

where:

EMA = exponential moving average

3) *Bollinger Band (BB) - Upper & Lower*

BB has plotted two standard deviations away from a simple moving average. Standard deviation is a measure of volatility when the markets become more volatile, the bands widen; during less volatile periods, the bands contract. It is believed that the closer the prices move to the upper band, the more overbought the market, and the closer the prices move to the lower band, the more oversold the market. Approximately 90% of price action occurs between the two bands [9].

$$\text{BOLU} = \text{MA}(\text{TP}, n) + m * \sigma[\text{TP}, n]$$
$$\text{BOLD} = \text{MA}(\text{TP}, n) - m * \sigma[\text{TP}, n]$$

where:

BOLU = Upper Bollinger Band
BOLD = Lower Bollinger Band
MA = Moving average
TP (typical price) = (High + Low + Close) ÷ 3
n = Number of days in smoothing period (typically 20)
m = Number of standard deviations (typically 2)
$\sigma[\text{TP}, n]$ = Standard Deviation over last n periods of TP

4) *Relative Strength Index (RSI)*

RSI is a momentum indicator that measures the magnitude of recent price changes to analyze overbought or oversold conditions. It is primarily used to attempt to identify overbought or oversold conditions in the trading of an asset. RSI values range from 0 to 100. Traditional interpretation and usage of the RSI are that RSI values of 70 or above indicate that the asset is becoming overbought or overvalued, and therefore may be primed for a trend reversal or corrective pullback in price. On the other side of RSI values, an RSI reading of 30 or below is

commonly interpreted as indicating an oversold or undervalued condition that may signal a trend change or corrective price reversal to the upside [10].

$$RSI_{\text{step one}} = 100 - \left[\frac{100}{1+\frac{\text{Average gain}}{\text{Average loss}}}\right]$$

$$RSI_{\text{step two}} = 100 - \left[\frac{100}{1+\frac{\text{Previous average gain}*13+\text{Current gain}}{\text{Average average loss}*13+\text{Current loss}}}\right]$$

5) *Money Flow Index (MFI)*

MFI is a momentum indicator that measures the inflow and outflow of money into security over a specific period of time. The MFI uses a stock's price and volume to measure trading pressure. Since MFI adds trading volume to the relative strength index (RSI), it is sometimes referred to as volume-weighted RSI [11].

$$\text{Money Flow Index} = 100 - \frac{100}{1 + \text{Money Flow Ratio}}$$

Where:

$$\text{Money Flow Ratio} = \frac{\text{14 Period Positive Money Flow}}{\text{14 Period Negative Money Flow}}$$

$$\text{Raw Money Flow} = \text{Typical Price} * \text{Volume}$$

$$\text{Typical Price} = \frac{(\text{High} + \text{Low} + \text{Close})}{3}$$

6) *Stochastic (STOCH)*

A stochastic oscillator is a momentum indicator comparing a particular closing price of a security to a range of its prices over a certain period of time. The sensitivity of the oscillator to market movements is reducible by adjusting that time period or by taking a moving average of the result. It is used to generate overbought and oversold trading signals, utilizing a 0-100 bounded range of

values. A value over 80 is considered overbought and a value below 20 is considered oversold [12].

$$\%K = \left(\frac{C - L14}{H14 - L14} \right) \times 100$$

where:

C = the most recent closing price;

L14 = the lowest price traded of the 14 previous trading sessions;

H14 = the highest price traded during the same 14-day period; and

%K = the current value of the stochastic indicator.

7) *Commodity Channel Index (CCI)*

The CCI compares the current price to an average price over a period of time. The indicator fluctuates above or below zero, moving into positive or negative territory. While most values, approximately 75%, will fall between -100 and +100, about 25% of the values will fall outside this range, indicating a lot of weakness or strength in the price movement [13].

$$CCI = \frac{\text{Typical Price} - \text{Moving Average}}{.015 \times \text{Mean Deviation}}$$

where:

$\text{Typical Price} = \sum_{i=1}^{P}((\text{High} + \text{Low} + \text{Close}) \div 3)$, where P = the number of periods

$\text{Moving Average} = (\sum_{i=1}^{P} \text{Typical Price}) \div P$

$\text{Mean Deviation} = (\sum_{i=1}^{P} \mid \text{Typical Price} - \text{Moving Average} \mid) \div P$

8) *Average Directional Index (ADX)*

Trading in the direction of a strong trend reduces risk and increases profit potential. The ADX is used to determine when the price is trending strongly. In many cases, it is the ultimate trend indicator. ADX is used to quantify trend strength. ADX calculations are based on a moving average of price range expansion over a given period of time. The best results are usually

obtained with 14 bars [14], although other time periods can be used. ADX can be used on any trading vehicle such as stocks, mutual funds, exchange-traded funds, and futures.

$$+\text{DI} = \left(\frac{\text{Smoothed } +\text{DM}}{\text{ATR}}\right) \times 100$$

$$\text{-DI} = \left(\frac{\text{Smoothed -DM}}{\text{ATR}}\right) \times 100$$

$$\text{DX} = \left(\frac{|\ +\text{DI} - \text{-DI}\ |}{|\ +\text{DI} + \text{-DI}\ |}\right) \times 100$$

$$\text{ADX} = \frac{(\text{Prior ADX} \times 13) + \text{Current ADX}}{14}$$

where:

$+\text{DM (Directional Movement)} = \text{Current High} - \text{Previous High}$

$\text{-DM} = \text{Previous Low} - \text{Current Low}$

$\text{Smoothed } +/\text{-DM} = \sum_{t=1}^{14} \text{DM} - ((\sum_{t=1}^{14} \text{DM}) \div 14) + \text{Current DM}$

$\text{ATR} = \text{Average True Range}$

9) *Moving Average Convergence Divergence (MACD)*

MACD is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-period EMA from the 12-period EMA. The result of that calculation is the MACD line. A nine-day EMA of the MACD called the "signal line," is then plotted on top of the MACD line, which can function as a trigger for buy and sell signals. Traders may buy the security when the MACD crosses above its signal line and sell - or short - the security when the MACD crosses below the signal line [15].

$$\text{MACD} = 12 \text{ period EMA - } 26 \text{ period EMA}$$

10) *William's %R (WILLR)*

Williams %R, also known as the Williams Percent Range, is a type of momentum indicator that moves between 0 and -100 and measures overbought and oversold levels. The Williams %R may be used to find entry and exit points in the market. It compares a stock's closing price to the high-low range over a specific period, typically 14 days or periods [16].

$$\text{Wiliams } \%R = \frac{\text{Highest High} - \text{Close}}{\text{Highest High} - \text{Lowest Low}}$$

where:

Highest High = Highest price in the lookback period, typically 14 days.

Close = Most recent closing price.

Lowest Low = Lowest price in the lookback period, typically 14 days.

11) *Awesome Oscillator (AO)*

The Awesome Oscillator is an indicator used to measure market momentum. AO calculates the difference of a 34 Period and 5 Period SMA. The SMAs that are used are not calculated using closing price but rather each bar's midpoints. AO is generally used to affirm trends or to anticipate possible reversals [17].

```
lengthAO1=input(5, minval=1) //5 periods

lengthAO2=input(34, minval=1) //34 periods

AO = sma((high+low)/2, lengthAO1) - sma((high+low)/2, lengthAO2)
```

12) *Balance of Power (BOP)*

The BOP is based on oscillators, labeling the components as "bull power" or "bear power." These are combined with an EMA, which is a trend-following indicator essential to the calculation. Bull power is a simple calculation, derived by subtracting an EMA (perhaps a 13-day EMA) of closing prices from a high price of any given security. Bear power subtracts the EMA from the corresponding low price of that trading day [18].

Bull Power = Period High − 13 Period EMA
Bear Power = Period Low − 13 Period EMA
where:
Period High and Period Low = High or low price for the time period used, such as a daily chart or a 1-hour chart.
EMA = Exponential Moving Average

THEORY

We used two algorithms to generate our predictions, Convolution Neural Network (CNN) and Long Short-Term Memory (LSTM). The fundamental problem is a combination of both classification (discreet) and regression (continuous). These two models have shown favorable results in previous works related to stock market predictions with good accuracy and thus proves to be a good candidate for Forex market prediction.

1) **Convolutional Neural Network -** is a regularized version of multilayer perceptron (MLP) and uses a feedforward artificial neural network generating a set of outputs from a set of inputs. CNN is characterized by several layers of input nodes connected as a directed graph between the input and output layers. CNN is a combination of hidden layers such as Convolutional Layers, Pooling Layers, Regularization Layers, and Fully Connected Layers. Convolutional layers are the main building blocks of CNN. Convolutional Layer is used to convolve one matrix to another by doing dot product with the filters, Pooling layers are used to pool the feature matrices by reducing the dimensions, Normalization layers normalize the matrices for faster training such as Batch Normalization, Regularization layers like Dropout layer will zeroed out some units of layer for avoiding overfitting issue and fully connected layers are used for classification task which is connected to the output layer. Mathematically it is a cross-correlation between two matrices [19].

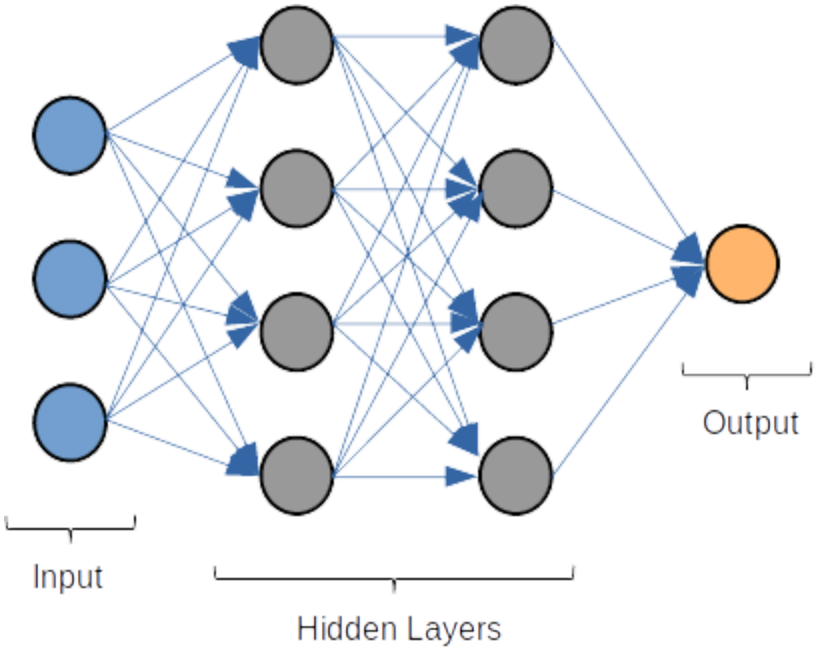

Fig: Neural Network with two hidden layers

We used ReLU as the activation function for this algorithm. It is actually called Rectified Linear Unit. The equation of ReLU is :

$$f(x) = max(x, 0)$$

Fig: Equation of ReLU function

If x < 0, the output of ReLU will be 0 otherwise it will give the raw input as output because it will choose the maximum of (x,0). It is widely used in deep learning to accurately predict results.

We have used softmax function in the output layer as the activation function. It gives the output of each unit between 0 and 1, just like a sigmoid function but it divides each unit such that the total sum of the units is equal 1. The output of the softmax function is a categorical probability distribution. The equation of softmax function is shown below, where 'z' is a vector of output

units, ‘j’ indexes the output units and ‘k’ is also used as index to sum all the units of the output vector.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Fig: Equation of Softmax function

2) **Long Short-Term Memory** - is an artificial recurrent neural network. Unlike standard feedforward neural networks, LSTM has feedback connections. LSTM is mainly used for time series problems where the output of a prediction affects the input of the next prediction. The choice of LSTM stems from the perspective of time-series prediction where an inherent correlation is present among the data. The key difference between traditional feed-forward Neural Networks and LSTM is the recurrence among the nodes used in LSTM. When data is input to an LSTM network, the flow of data is essentially controlled inside the cell (node) using "input", "output" and "forget" gates. Since these gates act as filters to what the cell receives and when it can receive, the nodes used inside these networks can remember values for an arbitrary period of time. To reduce error during training, the nodes receive values which transfer back after an activation function (output layer) i.e backpropagation [20].

The model that was created uses this concept of the relationships among data that is dependent as the series moves forward. For the LSTM network, the data was structured into 23 slices, where

the 24th slice is predicted using the model which then, in turn, acts as input for another round of training. This works synergistically as the algorithm works best if there is a relation among the data and is insensitive to time-gaps. Since the Forex market can have random events at different points in time which may trigger prices changes, traditional Recurrent Neural Networks can act worse since it is more sensitive to such events. RNNs suffer from "Vanishing" (close to zero) and "Exploding" (very high value) gradients during loss calculation and as such errors creep up and produce an unsatisfactory prediction whereas LSTM fares better in this regard.

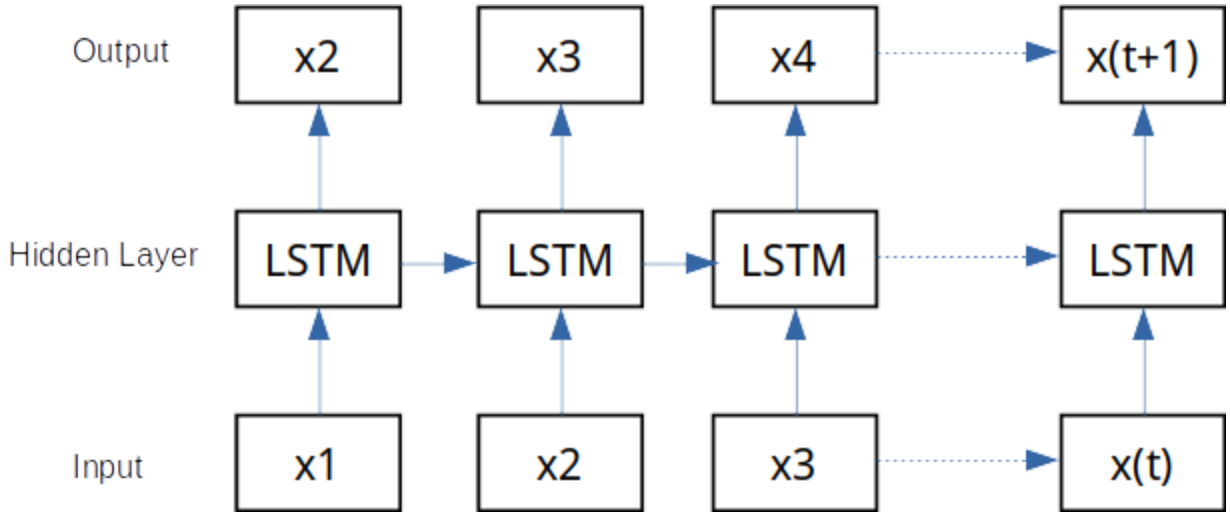

Fig: Overview of LSTM architecture.

3) **AutoRegressive Integrated Moving Average (ARIMA)** - is a form of regression analysis that gauges the strength of one dependent variable relative to other changing variables. The model's goal is to predict future securities or financial market moves by examining the differences between values in the series instead of through actual values [21].

An ARIMA model can be understood by outlining each of its components as follows:

- *Autoregression (AR)* refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.
- *Integrated (I)* represents the differencing of raw observations to allow for the time series to become stationary, i.e., data values are replaced by the difference between the data values and the previous values.
- *Moving average (MA)* incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each component functions as a parameter with standard notation. For ARIMA models, a standard notation would be ARIMA with p, d, and q, where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as:

- *p*: the number of lag observations in the model; also known as the lag order.
- *d*: the number of times that the raw observations are differenced; also known as the degree of differencing.
- q: the size of the moving average window; also known as the order of the moving average.

DATA PREPROCESSING

SET 1

Features: Open, High, Low, Close, Volume.

Output: Buy/Long or Do Nothing classification (1 = buy, 0 = nothing). Our condition to generate a buy signal was that within the next 30 minutes there should be at least a 10 pips movement towards the long side and less than 5 pips movement towards the short side. In this case, even a 50% accuracy will mean 100% Return On Equity (ROE) because our Profit:Loss target (ratio) is set to 2:1 for the entries.

Model Used: CNN

SET 2

Features: RSI, STOCH, CCI, ADX, AO, MOM, MACD, STOCH RSI, WILLR, BOP, ULTOSC.

Output: Same conditions as SET 1.

Model Used: CNN

SET 3

Features: Open

Output: Predict the next 24 hours Open prices based on previous 48 hours Open prices (1 hr time frame).

Model Used: LSTM

SET 4

Features: BB, CCI, MFI, RSI, PSAR, EMA, SMA

Output: Do Nothing, Buy and Sell (0, 1, 2 respectively) classifications. Condition to generate the signals were as follows:

> *Sell Point = 0;*
>
> *Buy Point = 0;*
>
> *If the current candle's closing price < the next 12 candle's closing price.*
>
> *Then Sell Point = Pips difference between the current candle's closing and the lowest closing price between the next 12 candles.*
>
> *If the current candle's closing price > the next 12 candle's closing price.*
>
> *Then Buy Point = Pips difference between the current candle's closing and the highest closing price between the next 12 candles.*
>
> *If Sell Point > Buy Point Then Sell (2)*
>
> *ElseIf Sell Point < Buy Point Then Buy (1)*
>
> *Else Do Nothing (0)*

Model Used: CNN

SET 5

Features: Open

Output: Predict the next 24hrs Open prices based on previous 48 hours Open prices (1hr time frame).

Model Used: ARIMA

METHODOLOGY

For SET 1, SET 2 and SET 4 we used CNN where 80% of the data as training set, 10% of the data as validation set and 10% of the data as test set for cross-validation. Then we performed one hot encoding on the data that was split. Then all the data was reshaped and fed into the CNN model for training. We used two blocks of Conv1D layer which had a kernel size of 3 and 64 filters. Then we added a Dropout layer (30%). After that, we flattened the matrix and passed into a fully connected layer of 100 units. Lastly, we added an output layer of 2 units with softmax activation. This is because we needed a probabilistic distribution of the output. We used 'categorical cross-entropy' as our loss function, 'Adam' as the optimizer.

For SET 3 we used LSTM where the "Open" data was used as the training set. Then we structured it into sets of 24 data entries and 1 output. Then this data was reshaped and feed into the LSTM model. We used the LSTM layer with 50 units and then we added a Dropout layer (20%) after which we added another LSTM layer of 50 units and a Dropout layer (20%). Lastly, we added a Dense layer with one unit which is our output layer. We used 'mean squared error' as our loss and 'Adam' as the optimizer.

SET 5 was prepared to make a comparison with LSTM. For simplicity, we have used an add-in from real-statistics.com with Microsoft Excel to directly integrate ARIMA into our workbook. For this model lag order was set to 1, degree of differencing to 0 and order of moving average to 0.

PROGRAMMING LANGUAGES AND ENVIRONMENT

Data preprocessing:

- Numpy (python library) (https://docs.scipy.org/doc/numpy/reference/)
- Pandas (python library) (https://pandas.pydata.org/pandas-docs/stable/reference/index.html)
- Microsoft Excel

Model building:

- Keras framework (python) (https://keras.io/documentation/)
- Real Statistics Resource Pack (Microsoft Excel Add-in) (http://www.real-statistics.com/)

Programming Languages:

- Python
- Visual Basic for Applications (VBA)

IDE:

- Jupyter notebook (https://jupyter-notebook.readthedocs.io/en/stable/)

Execution Platform:

- Google Colab (https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/01.01-Help-And-Documentation.ipynb)

RESULTS

We have applied CNN model on SET 1 and SET 2 accordingly as mentioned in the data preprocessing section. The following results were obtained:

SET 1	CNN		SET 2	CNN
Training Accuracy	0.8735		Training Accuracy	0.8734
Training Loss	0.3548		Training Loss	0.3799
Testing Accuracy	0.8731		Testing Accuracy	0.8727
Testing Loss	0.3547		Testing Loss	0.3813

Table 1. Results of SET 1 & SET 2

We have noticed that the accuracy was around 87.3% because the total numbers of 0s were also 87.3%. So the model basically generated 0 for any input which was thrown at it. The reason for this was imbalanced dataset. Balancing the data would mean removing out a significant number of rows containing 0s but it was those candlesticks in the sequence which resulted in the formation of the candlesticks which had 1s as their output. Therefore, there was no solution to fix this balancing problem because we could not just remove chunks of data out of a sequence.

For SET 3 we have a set of previous 48 hours OPEN price data was feed into the LSTM model and it predicted the next 24 hours OPEN prices. The result of the test is shown in the chart below.

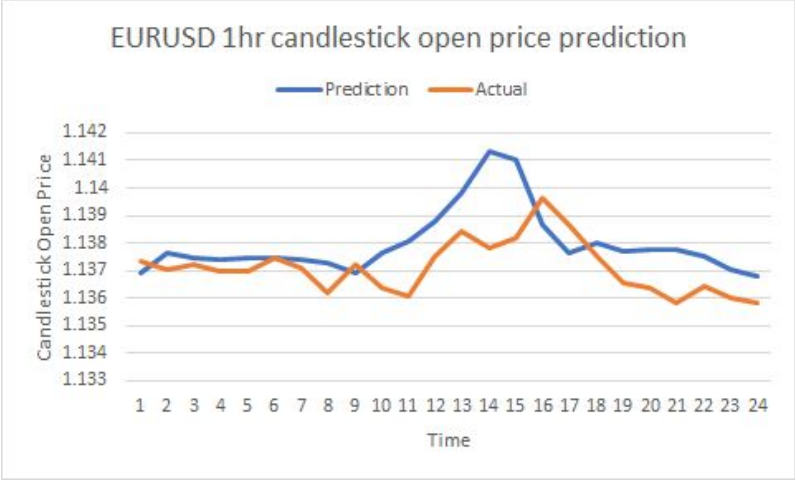

Fig 2. LSTM prediction vs actual open price

For SET 4 we have faced similar problems as SET 1 and SET 2. In this case, the number of 0s was roughly around 73% and our accuracy being near that value also suggests that it predicted 0 for any input thrown at it.

For SET 5 the same data as LSTM was input to ARIMA to predict the same output as that of LSTM in SET 3. The result of the test is shown in the chart below.

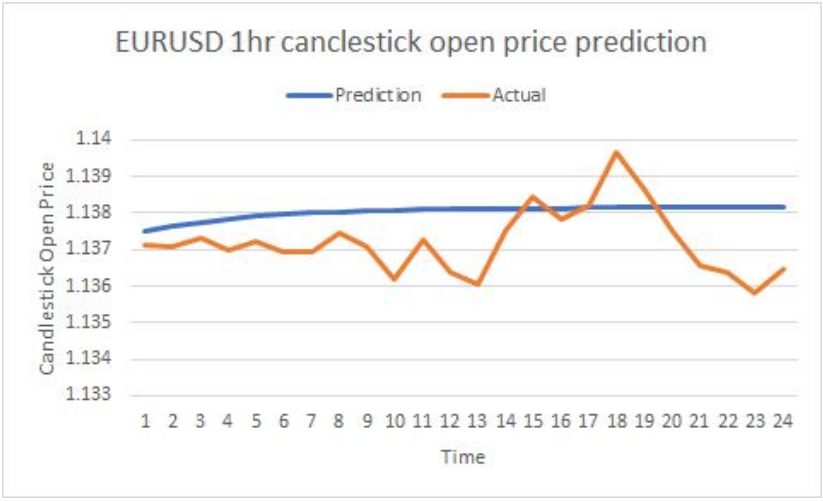

Fig 3. ARIMA prediction vs actual open price

Our reasoning behind such a prediction from ARIMA was that since it uses moving average, it relies heavily on the past candlesticks data rather than the latest ones which causes it to weight down more on past trends rather than latest trends. To validate our reasoning we have performed another test where EURUSD was the most volatile in 2018, which was in March. The results showed that ARIMA stuck with the past trend which was bullish (upwards) to predict the new candlestick open prices. This suggests that AIRMA should work better with higher timeframe such as daily or weekly candlesticks because on those kinds of time frames the overall trend doesn't drastically change as much as they do on smaller time frames such as hourly or minutes.

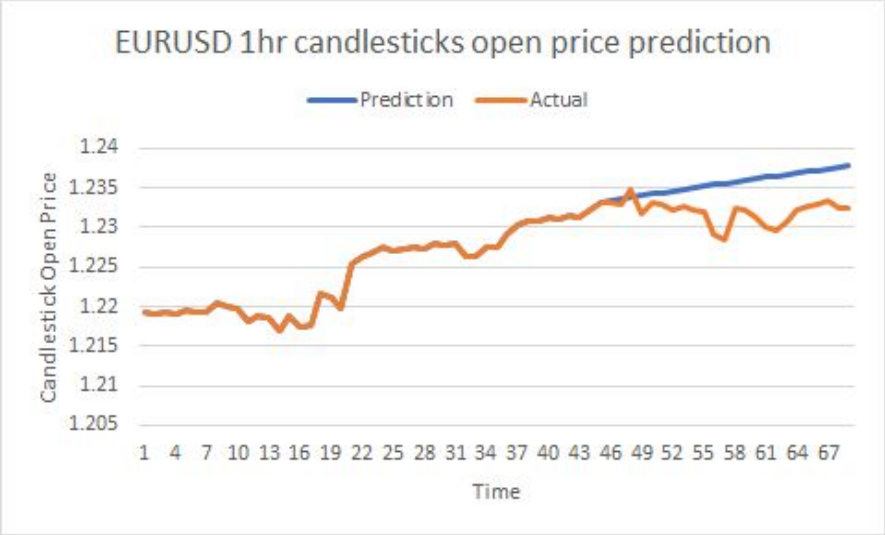

Fig 4. Test result to validate our reasoning behind ARIMA's prediction

CONCLUSION

We used two different approaches, to predict the opening price itself and secondly prediction stances for taking entries or not. This facilitates the decision of asking if a position should be opened or not, and if it should open then at which side (i.e. long or short) and at which price should we set our Take Profit (TP) and Stop Loss (SL). The problem we faced with SET 1, SET 2 & SET 4's CNN classification was that the model was predicting everything as 0 (i.e. do not take any position) because the number of 0s in the set was the majority (almost 6.7 folds higher than 1s), therefore, the algorithm only learned to predict everything as 0. It was not possible to reduce the number of 0s to render a better prediction for a favorable outcome because removing entries from the middle of a time series data set affects those events. These events play a role to create the next set of inputs. We plan on solving this problem with CNN by trying to predict trends instead of opening positions or not which should work better for classification problems as this data imbalance will no longer be an issue. With SET 3 we did not face any problems and the loss was very low. But, while comparing SET 3 versus SET 5 results which is LSTM against ARIMA respectively, we have observed that ARIMA has performed very poorly compared to LSTM. The reason behind this is because ARIMA mainly depends on moving average which weight down on past candlesticks data more than the latest candlesticks which cause it to react very slowly to latest trend changes in the market. The main reason for utilizing two different approaches was to use one model to predict the trend of the market while the other will provide us with a probable target to exit the market.

FUTURE WORK

We plan on extending the LSTM model to predict OPEN, HIGH, LOW, CLOSE and combine them to predict a full candlestick which will be used for determining the Stop Loss (SL) and Take Profit (TP) dynamically to get the best possible outcomes for an open trade position. Determining entry points in our classification models failed due to imbalanced data, but we will try to determine market trends instead using classification techniques. The imbalance issue should not raise here because trend movements are in abundance for any time period for any given currency pairs. By utilizing the prediction of market trend and price, we plan on creating entry and exit point signals in real time. The end goal of this project is to build an auto trading bot which will analyze the market conditions and act upon feedbacks and decisions from the deep learning models to make and auto adjust trade decisions in real time.

REFERENCES

[1] Bank.S, Khan.K & Anwer.M. *Hybrid Machine Learning Technique for Forecasting Dhaka Stock Market Timing Decisions*. Retrieved from Computational Intelligence and Neuroscience,feb 19. 2014

[2] Pate.J, Shah.S & Thakkar.P .*Predicting stock and stock price index movement using Trend 4 Deterministic Data Preparation and machine learning techniques*. Retrieved from Computer Science & Engineering Department, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India, 2014

[3] Xu.B, Zhang.D, Zhang.S, Li.H, and Lin.H, *Stock Market Trend Prediction Using Recurrent Convolutional Neural Networks.Dalian University of Technology, Dalian, China*

[4] Huy D. Huynh.H, Dang.L.M and Duong.D *A New Model for Stock Price Movements Prediction Using Deep Neural Network*

[5] Dukascopy, *Historical Data Export*.https://dukascopy.com/plugins/fxMarketWatch/?historical_data

[6] Dukascopy,*Awards*.https://www.dukascopy.com/swiss/english/about/awards/

[7] Investopedia,*Terms*.https://www.investopedia.com/terms/s/sma.asp

[8] Investopedia,*Terms*.https://www.investopedia.com/terms/e/ema.asp

[9] Investopedia,*Terms*.https://www.investopedia.com/terms/b/bollingerbands.asp

[10] Investopedia,*Terms*.https://www.investopedia.com/terms/r/rsi.asp

[11] Investopedia,*Terms*.https://www.investopedia.com/terms/m/mfi.asp

[12] Investopedia,*Terms*.https://www.investopedia.com/terms/s/stochasticoscillator.asp

[13] Investopedia,*Terms*.https://www.investopedia.com/terms/s/commoditychannelindex.asp

[14] Investopedia,*Terms*.https://www.investopedia.com/terms/a/adx.asp

[15] Investopedia,*Terms*.https://www.investopedia.com/terms/m/macd.asp

[16] Investopedia,*Terms*.https://www.investopedia.com/terms/w/williamsr.asp

[17] Tradingview,*Wiki*.https://www.tradingview.com/wiki/Awesome_Oscillator_(AO)

[18] Investopedia,*Terms*.https://www.investopedia.com/terms/e/elderray.asp

[19] NVIDIA,*NVIDIA Developer*.https://developer.nvidia.com/discover/convolutional-neural-network

[20] Skymind,*Wiki*.https://skymind.ai/wiki/lstm

[21] Investopedia,*Terms*.https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp