Using Neural Networks to Predict Forex Prices

Vincent Wang

March 14, 2019

Contents

Part I

Analysis

1 Research

1.1 Forex Trading

The Forex (foreign exchange) trading market is huge. Every day, $5.3 trillion US Dollars are traded on the forex market - 53 x the volume that is traded on the New York Stock exchange [**?**].

Successful traders are often those that have lots of experience with markets. Over time they gain some intuition or "feel" for how the market will act. That being said however, markets move randomly. Trading, especially forex trading, has been likened to gambling because of this - it's risky and very difficult to reliably predict [**?**]. Even when a correct prediction is made, margins in forex are very small as the markets do not move a large amount so turning a profit is difficult, especially when taking into account the broker's fees to carry out the trade. In order to make any significant profits, large investments need to be made, which carries large risk with it.

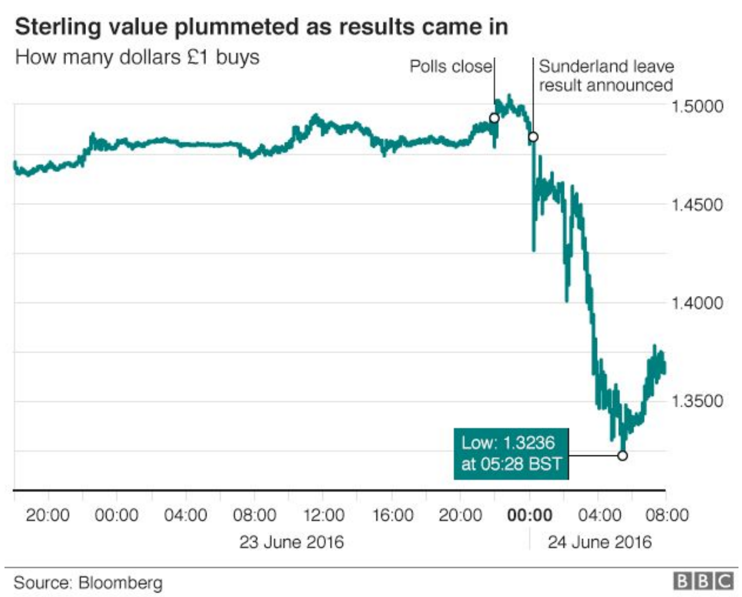

Figure 1: Pound against the dollar around the result of the Brexit Vote [**?**]

To an extent, a trader can try to predict long term forex trends through following current events. For example, if a country is going through a period of political instability or uncertainty, a trader might choose to sell that currency. For example, the result of the 2016 "Brexit" vote caused the pound to fall to a 31 year low [**?**] *(See Figure 1).* This aspect of trading strategy presents challenges for algorithmic trading as it is difficult to inform a program about the political climate of a country.

1.1.1 Mechanics of Trading

When trading forex, we talk about trading currency pairs [**?**]. A currency pair is represented in the form *base currency/quote currency* and its value is how much of the quote currency the base currency buys. For example, EUR/USD = 1.2500 means that the Euro buys 1.25 US Dollars. When a pair is bought or sold, it involves buying ("going long") on one currency while simultaneously selling ("going short") on the other. E.g. putting a buy order on EUR/USD means going long on Euros while going short on Dollars.

Every time a currency pair is traded on the forex market, its price changes. If the price rises in a certain time frame the market described as bullish, if it falls it is described as bearish. The goal of a forex trader is to to try predict these changes and open buy or sell orders so that they can make a profit off the market movement when the order is closed. For example, if a trader thought that the price of EUR/USD was going to rise (the Euro strengthens against the dollar), they would buy EUR/USD. If the price of the Euro rises against the dollar, closing the trade at that point (selling the Euros) would result in an overall profit - the of Euros bought when closing the trade are more valuable than when the trade is opened, and so is worth more in whatever currency that the account is denominated in. If a downwards movement is predicted, then one can sell at the higher price and buy back at a lower price to make a profit.

For day traders (the group for which this project is aimed at) one needs to open an account with a brokerage firm. An account is opened with a base currency - the currency with which is used to buy/sell assets and the currency which profits/losses are given in. To carry out trades, one deposits money into their account with the broker. Brokers can make money in a number of different ways, including putting commission on trades (which is discussed below) or buy offering services such as data analysis or advice for a monthly fee.

1.1.2 Buying vs Selling

In trading, one does not have to previously own an asset to sell it. Shorting a currency works by borrowing the specified amount of the currency from your broker, agreeing to buy it back in the future at the future price.

Because of this, shorting has an inherent difference to buying in forex. When buying, you are betting on the price of a currency pair to rise. The worst outcome of a trade is that the value of your trade goes to 0 as the price of the pair does i.e. you lose all of your initial investment. When shorting, there is theoretically no limit on how much you could lose. The price of a currency pair can keep rising, and with it the amount you need to repay when closing the trade does [**?**]. In practice, the value of a pair will not keep rising to infinity, however, it is important to consider this when shorting.

To protect against the danger of this, you can set a stop order when you start a trade. There are discussed more below.

1.1.3 Broker Fees

Broker fees can come in a number of different varieties. Disregarding fees a broker might charge for advice or other services, two common fee types are spread and commission [**?**].

Spread is the difference between the buy and sell price of a currency pair quoted by a broker. This difference is given in pips - the fourth decimal place of a quoted price.[1] Spreads can be fixed or variable depending on the broker. Variable spreads could depend on market volatility or trading volume for example - if a currency pair moves a large amount, a broker would prefer to set a larger spread.

Commission can come in fixed and variable forms as well. Fixed fee commissions tend to be very large, and targeted for people trading at high volumes. Variable fees are dependant on the volume traded, and so offer a good middle ground for all traders. Because of this, variable fees are growing in popularity [**?**].

1.1.4 Trading on Leverage

As discussed above, currencies on the forex market do not move large amounts, and so huge investments are needed to make any non insignificant profits. To help with this individuals can trade with leverage from their broker.

Trading on leverage is the act of borrowing money to boost the size of an investment. It acts as a multiplier on the original investment, increasing both the potential profits and losses from it. For example, if 100 worth of USD is bought with 50:1 leverage, the trade has a value of 5000. If the trade is closed when GBP/USD has moved up 20 pips, 10 is made instead of the 0.20 made if leverage is not used. [**?**]This also works the opposite way however - if GBP/USD moves down 20 pips, then 10 is lost. Again, we can use stop orders to help protect against the risks of this.

[1] Imagine you are trading EUR/USD. The charts show a value of 1.2000 however your broker quotes two prices - a buy price of 1.2002 and a sell price of 1.2000. In this case we would say the spread is 2 pips. The small percentage on top of the actual value of the currency for the buy price is how the broker can make money off the spread.

1.1.5 Stop Orders

Stop orders can be used by traders to decrease the chance of loss on a trade. Stop orders are instructions for a broker, telling them to close a certain trade when the market has reached a certain price. Stop orders can be used to both minimize losses and secure profits. e.g. if after buying a currency pair, the value starts to decrease, a stop order lower than the initial value can protect from the initial investment depreciating in value too much. On the other side, if a trader is buying a pair and is happy to "cash out" when they have made a certain amount off the market movement, they can set a stop order with a price higher than that bought, to protect from negative impact of potential future downward movements. Stop orders could be used for volatile markets, or on trades that won't be able to be appropriately monitored by the trader. [**?**]

1.1.6 "2% Rule"

Stop orders can be used to help follow the 2% rule - a strategy used to balance risk and reward in which a trader risks no more than 2% of their available capital between all trades at any one time. When creating stop orders (and trading on leverage) one might use this to decide on the price to set the stop order at. [**?**]

1.1.7 Open High Low Close

When, trading it can be useful to look at other meta data in addition to the raw exchange rate to inform what action to take. For example, if there is an indication that the market is very volatile at a point in time, one might wish to hold off on a trade as a prediction could be more likely to be false.

Open High Low Close data gives us some indication of the volatility of an asset during a particular time frame. Open/Close prices are the price of the asset at the start/end of the time period. High/Low are the greatest and smallest prices within a time frame. [**?**]

These charts are typically represented in two ways. The first is simple known as a "bar chart". It has two short horizontal dashes - one line pointing to the left (back in time) at the opening price and one line pointing to the right (forward in time) indicating closing price. The range (high/low) is given by a vertical line with one end at the highest price and another at the low [**?**]. *(See Figure 2)*

The other is known as a "Japanese Candlestick Chart" a thin vertical line represents the range, and a thicker vertical box represents the open and close price [**?**]. *(See Figure 3)*

Figure 2: Example of an OHLC Bar Chart [**?**]

Figure 3: Example of an OHLC Japanese Candlestick Chart [**?**]

Both of these graphs (especially Japanese Candlestick) are usually colour-coded to help distinguish between bullish and bearish movements.

1.1.8 Why Trade Forex?

Forex trading attracts people for different reasons. One thing that makes forex attractive is that because movements are small, leverages offered are much larger than those on the stock market. This allows people to start trading effectively with relatively small sums of money. Unlike the stock market, forex is also open 24 hours a day (although retail brokers close on weekends).

Others however might not trade forex with the intention of directly making money. If a trader was trading US stocks, they might be worried about the potential decline/volatility of the dollar. To offset this - allowing them to still make money off their stock trades overall, they could short USD against Euro[**?**]

2 Proposed Solution

The solution should be an aide to a day trader, giving short term (intraday) predictions in 15 minute intervals. It will take in data from previous prices in the market, and give some form of prediction of price movements for EUR/USD for a number of different points in the near future (e.g. 30 mins, 1 hour, 2 hours etc.). Neural networks will be used to make these predictions.

The solution should include a web frontend that displays predictions along with measures of the accuracy of predictions in a graphical form to give users more information with which to make a judgement. This will include an api, with which a user should be able to get all data shown on the web page at a point in time in numerical form to be able to manipulate however they desire.

Originally it was thought the solution would be one which carries out trades directly, however this was thought to be both too difficult as well as too limited - directly carrying out trades would require committing to a single api broker. In addition, it was thought that there would not be anyone who would be willing to use it as it would mean entrusting one's assets to a service that's out of they don't have much control of.

3 Specification

//do Specification table here

3.1 Justification

Justification

Part II

Design

4 Networks

4.1 Data in/out

Each model should be able to work with 4 points of data at every time unit: open, high, low, close, or less. This is to fit the structure of incoming/real-world data provided by the AlphaVantage api (FX Intraday).

To train/test our agents, a dataset of open/high/low/close/volume from a six year period on the EURUSD market with 15 min intervals will be used. We will start tests using an 80/20 split of training data to test data, as this is a good balance between ensuring that the agent learns enough during training and accurately assessing the agent's ability.

Neural networks work better when inputs are normalised i.e. take on values between -1, and 1. To achieve this (or at least a proxy for this), raw input values were changed to ***(actual value - the mean of close prices of the window) * 100*** (the window is the time range for which the network is given inputs) Even for large windows, inputs to the network would very rarely exceed 1 or -1.

For outputs the inital plan was to let the network choose from one of three options - the price at a timestep will be within the spread of, or greater/lower than the spread of the most recent timestep. Each option would have a confidence level - the percentage chance the network attributes to that outcome.

4.2 Proposed Networks

We have two problems to consider with the how inputs to the network. Firstly we need the network to be able to see many previous time steps as this is what will likely allow it to predict prices correctly if anything. Secondly, for each time step, we have 4 data points - open, high, low, close.

A variety of input arrangements were proposed for the networks.

- **Feed-forward Networks:** All the feed-forward networks below would have a 1D convolutional layer as the first input layer, convolving each of the four inputs for a timestep (open, high, low, close) and producing one output for the next layer

- **Normal deep learning network:** The network would be made up of standard feed-forward connections in which every ouput in one layer would be an input to every node in the next layer. While straightforward to implement, this requires many neurons to train
- **Casually dilated convolutions:** This technique was inspired by Deepmind's wavenet [**?**]. It has the advantage of being able to look back on many previous timesteps while having a relatively small number of neurons to train. The structure of the network has the shape of a binary tree, where each layer has half the neurons of the last and each neuron takes two inputs from the previous layer. This allows the network to have a large lookback period without requiring many neurons.
- **Exponetially increasing merged timesteps:** This setup is motivated by the assumption that the further back in the past the data is, the less relevant it is. For this, we will take current set of 15 min data (0 to -15), the set of data from -15 to -30, then -30 to -60, -60 to -120, -120 to -240 and so on (if needed). To merge timesteps, we take the open value of the timestep furthest back in the past for the window we are looking at, the close from the most recent, and the maximum high value and minimum low value throughout the range.

- **Recurrent Networks:** Recurrent networks are good at solving problems that have a sequential nature, such as time series problems. Because of this, they are very appropriate for the problem at hand, however they take longer to train [**?**].
  - **LSTMs**: LSTM (Long-Term Short Memory) networks are especially good at solving sequenced problems as each lstm unit in the network chooses to remember of forget certain values depending on how important the network deems the value to be.

It was decided that a feed-forward network with "normal inputs" and a recurrent network should be tested before the other feed-forward networks as they would be more difficult to implement

4.3 Training Process

4.3.1 First Network

The initial model was a normal deep learning network. N previous sets of OHLC data were fed to the network and 3 outputs were given, representing the probability the network gave to the price moving up, down and staying within the typical spread price.

As discussed above, the first layer was a 1D convolution that created one output for each set of OHLC data. For the other hidden layers, a number of setups were tested, with usually a starting layer of 64 neurons. The output layer had three neurons with a softmax [2] applied so the network outputs represented the probability the network assigned to the price moving up, down, or staying within the spread. The network was initially setup with the SGD optimiser, with a small learning rate of 0.01 as is customary for deep supervised learning.

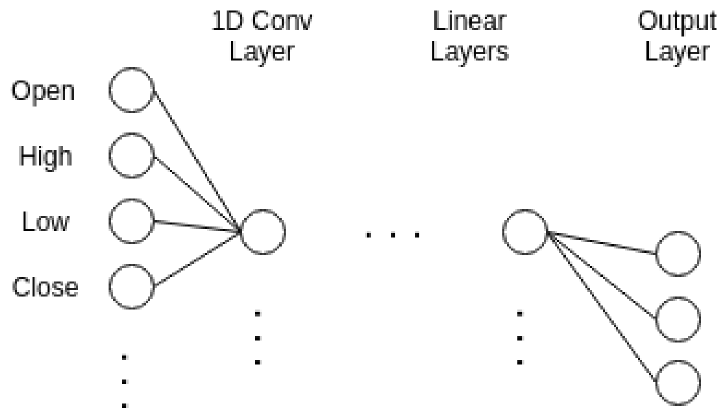

Figure 4: Network Diagram

As a preliminary test, it was decided the network should try to overfit [3] on a small batch first. The network greatly struggled with this however. On every run the network would either converge on a solution that assigns equal probability to each outcome, or one that always assigns 100% to the price increasing or decreasing.

Eventually, using the Adam optimiser instead of SGD, the network managed to overfit on the small batch of 20.[4]

It was thought that the reason Adam worked was because the problem is quite complex and thus SGD (steepest gradient descent - which takes steps only in the direction of the steepest gradient) was less likely to have found the global minimum of the cost function, whereas Adam, which is stochastic, is better at exploring the landscape of the cost function and so more likely to find the global minimum.

When attempting the actual training (using the entire training dataset), all

[2] A softmax is a function that normalises a number of values. This is often used to get probabilities from an output layer.

[3] Overfitting is where a network learns the test data and corresponding outputs it is given instead of being able to generalise and learn which features of the inputs cause certain outputs, thus performing very badly on unseen data. This usually happens when the size of the test data is too small

[4] In figure 5 the rows above show the output tensors from the network for the last few inputs. Below the expected outputs vs. the network ouput with the largest value (the network prediction)

```
        [[9.9892e-01, 9.8515e-04, 9.6589e-05]],

        [[1.6991e-03, 9.9578e-01, 2.5194e-03]],

        [[6.1960e-06, 1.3911e-03, 9.9860e-01]],

        [[9.9898e-01, 6.5962e-05, 9.5714e-04]],

        [[9.9802e-01, 8.5546e-04, 1.1289e-03]],

        [[1.6151e-03, 7.7030e-04, 9.9761e-01]],

        [[1.3650e-03, 9.9714e-01, 1.4904e-03]],

        [[9.9772e-01, 1.7278e-03, 5.5671e-04]],

        [[1.3717e-06, 2.2189e-04, 9.9978e-01]]], grad_fn=<SoftmaxBackward>)

target   tensor([[0, 2, 2, 2, 0, 0, 0, 2, 2, 0, 0, 0, 1, 2, 0, 0, 2, 1, 0, 2]])
pred.    tensor([[0, 2, 2, 2, 0, 0, 0, 2, 2, 0, 0, 0, 1, 2, 0, 0, 2, 1, 0, 2]])
tensor(1.4444e-06, grad_fn=<MseLossBackward>)
```

Figure 5: Output from running the overfitting test.

runs gave unsatisfactory results. At best networks gave the correct prediction around 49% of the time. Given that at the 15 minute interval the price moves up/down around 45% of the time and stays within the spread the remaining 10%, the network was not giving desirable predictions (the network was barely doing better than random guesses).

The approach that was taken to this problem of mapping the inputs onto one of three discrete outputs was influenced by classic classification problems such as recognising images of handwritten digits in which there is a clear "correct" mapping between the inputs onto one output. However forex prices are stochastic - it is possible to determine with certainty given a set up of inputs what the "correct" output should be so it was thought that training a network using the three discrete outputs (price up, down, same) represented as a [1, 0, 0], was causing problems during optimisation.

It was decided that a network with a different structure should be tested.

4.3.2 LSTM Network

It was decided that the next test should be an LSTM network that predicted the price itself. N individual sets of OHLC values would be fed one by one into the network with one LSTM layer and one single output neuron. The Adam optimiser was used with learning rates from 0.05-0.4.

The network produced what were thought to be far more desirable results, with initial tests able to predict the price in 15 mins correctly within the spread (taken as 0.7 pips) roughly 60% of the time . This was initially surprising as originally, it was initially thought that predicting the price itself was a more difficult problem than just predicting the discrete movement of the price.

//predictions here

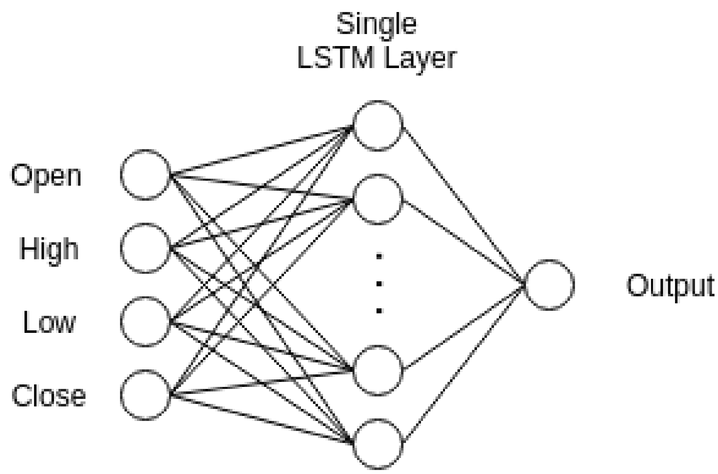

Figure 6: Network Diagram

The concern was that the network seemed to be just quoting the most recent close price at each (see **).

There were a few suggestions about why this might have been the case and how to improve on it.

- **Too few neurons** The network was not large enough and so couldn't learn the more complex behaviours required.
- **Window size was too large** Too much data was being fed to the network. With so many inputs it was difficult for the network to converge on a valid solution other than quoting the latest close price. Additionally, network outputs were given as movements from the mean of the window so the larger the window, the larger the window, the greater the variance of the mean in relation to the final close price, which could be affecting the validity of the predictions.
- **Prediction time-scale too small given the data** The tests were first being done on 15 and 30 minute predictions. Given the data being fed to the network was in 15 minute intervals, it was thought that these predictions could have been too close in the future for the network to converge on effective solutions as the price would not have moved too far/any movements had too much associated noise. It was thought that over longer time-scales general trends would be able to be found and thus more effective predictions could be made.

It was found that adding more neurons could often result in a worse solution, likely because the network was too complex and the number of training iterations was too small to be able to properly explore the landscape of the fitness function. When decreasing the window size, it was found that the network could converge on smaller losses, and when doing tests on the predictions further in the future, returned values were thought to be better.

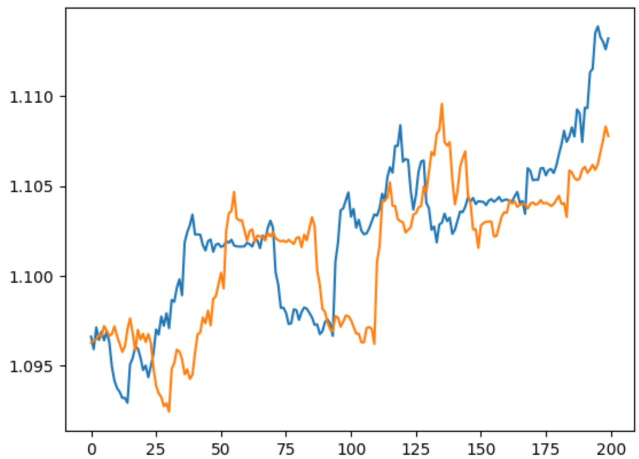

Figure 7: Sample prices against the prediction made that timestep 4 hours (16 timesteps) beforehand

In figure 7 while it does seem that the network mostly seems to be quoting the most recent close price, at many points we can see that it is both over and undershooting peaks and troughs - showing that it is observing trends and making decisions based on data from more than just the last timestep.

It was decided that this structure should be used in the final solution.

4.4 Measuring Success

Now it had been decided that a raw price prediction network would be used, the ability of a network was measured by two criteria - the mean squared error between the predictions and the target values, and the percentage of predictions made that fell within the high and low prices of the relevant timestep.

As discussed above, it seemed as though the network could have been just quoting the most recent close price as the prediction. To test this - ensure the network was making "valid" predictions, a baseline was created by finding the MSE and percentage accuracy of a solution quotes the most recent close price as the prediction for a future timestep.

4.5 Final networks

Using the above critera, the networks that were chosen to be used had the following results ***yeet a table of the successes in***

‘’

5 User Interface

5.1 Layout

The predictions should be shown in a concise way graphically so that users viewing predictions via the webpage can more easily interpret the data. All data should be shown on one page.

It was thought that previous market data, predictions as well as some measure of the precision of the predictions such as standard deviation of error should be displayed on one graph. If possible users should be able to choose how much data is being displayed at one time using sliders. Aon the same page, there should be a chart displaying the recent accuracy of predictions (how often the network correctly predicts the price within the actual high and low of the timestep).

On the page the current UTC time and the time of the data being shown should be displayed.

6 API

6.1 Data Returned

When an API call is made, the server will check if the API key used is valid (is in the list of API keys being used). If it is, then data will be returned, else an error message will be returned.

Data returned should be in JSON format and include all data that is shown on the webpage in numerical form. JSON was chosen as it is relatively compact, well supported and also allows for more logical represetation and ordering of data with the use of hierarchy.

A request should not be served if requests from the user are being made too frequently.

6.2 Signing Up

On the webpage, there should be a text field where a user can enter there email to get an API key. If the email is valid, a random API key will be generated and sent to the user. If the email is invalid an error message should appear. If the email has already been used to sign up, the API key should be shown.

6.3 Email Hashing

It is thought that the users would never need to be directly contacted and thus the user emails should never be stored or sent in plaintext to ensure user privacy. Were emails sent in plaintext, a "man in the middle" would be able to see the email.

To ensure this when a user signs up for the service, their email should be hashed on the client side using MD5 with a random salt before being sent to the server. The user will then be shown their API key on the webpage.

7 Backend

7.1 Dataflow

There should only be one copy of historic data and predictions to ensure data is consistent in all places and updating these is easier. Because of this, the same datafile that can be accessed via the API should be used as the source of data for the webpage.

7.1.1 Updating data

Every 15 minutes, the server should check for new data from AlphaVantage. Data from the API does not update every 15 minutes on the dot however so the server should keep making API calls until the time of the data changes. AlphaVantage stops fufilling requests if they exceed more than 5 requests a minute or more than 500 requests a day. Given that new data for each 15 minute timestep seemed to come roughly 5 minutes late, a 45 second delay between requests was thought to be appropriate.

7.1.2 Feeding data to networks

7.2 Database

The database should be made up of a User table, storing the a hashed email and api key, and a Predictions table, storing the time of the prediction and the value of each prediction. These tables have a many to many relationship and so need a helper table.

Tables required

- **User:** Stores Email hash, API key, Date of signing up.
- **Predictions:** Stores the time the prediction corresponds to and each prediction.
- **Requests:** Stores User ID, Prediction Time, current time. A helper table for the many to many relationship between the User and Predictions table. Every time a user makes an API request, an entry should be made for this table. This will allow for policing of users making requests too frequently. It will also allow for metrics such as when in the day requests are most frequently made to be calculated.

Part III
Final Design

Part IV
Testing

Part V
Evaluation

References

[] *50 Forex & Trading Industry Statistics & Trends From 2018* [2018].
URL: *https://brokernotes.co/forex-trading-industry-statistics/*

[] *Candlestick Chart* [n.d.].
URL: *https://datavizcatalogue.com/methods/candlestick$_c$hart.html*

[] Hannah, F. [2017], 'Beginner's guide to currency trading'.
URL: *https://www.independent.co.uk/money/beginners-guide-to-currency-trading-a7568956.html*

[] Lioudis, N. K. [2018*a*], 'Forex leverage: A double-edged sword'.
URL: *https://www.investopedia.com/articles/forex/07/forex$_l$everage.asp*

[] Lioudis, N. K. [2018*b*], 'Forex trading: A beginner's guide'.
URL: *https://www.investopedia.com/articles/forex/11/why-trade-forex.asp*

[] LLC, S. L. C. M. [2016], 'Calcuating forex costs in trading'.
URL: *https://www.youtube.com/watch?v=gt5sQN19VAk*

[] Momoh, O. [2018], 'Stop order'.
URL: *https://www.investopedia.com/terms/s/stoporder.asp*

[] *Open-high-low-close Chart* [n.d.].
URL: *https://datavizcatalogue.com/methods/OHLC$_c$hart.html*

[] *Pound plunges after Leave vote* [2016].
URL: *https://www.bbc.co.uk/news/business-36611512*

[] Russell, J. [n.d.], 'Learn about short selling currency and how it works in forex market'.
URL: *https://www.thebalance.com/what-it-means-to-go-short-in-investment-terms-1344960*

[] Staff, I. [2017], '2
URL: *https://www.investopedia.com/terms/t/two-percent-rule.asp*

[] Staff, I. [2018], 'Ohlc chart'.
URL: *https://www.investopedia.com/terms/o/ohlcchart.asp*

[] *The cost of trading forex* [n.d.].
URL: *https://learn.tradimo.com/dont-go-broke-protect-your-capital/the-cost-of-trading-forex*

[] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. and Kavukcuoglu, K. [2016], 'WaveNet: A Generative Model for Raw Audio', *arXiv e-prints* p. arXiv:1609.03499.