Kaiko Cryptocurrency Challenge

An Investigation into the predictability of arbitrage events
in the cryptocurrency markets.

Team: Student Ensemble

February 2021

Introduction

This report attempted to find profitable trading strategies in the cryptocurrency markets. While most trading strategies focus on buying or selling an asset and holding onto it until it appreciates or depreciates to a certain level, our trading strategy of interest was arbitrage. An arbitrage event happens in the crypotcurrency markets when the minimum ask price (the least someone is willing to accept for a cryptocurrency) on an exchange is larger than the maximum bid price (the most someone is willing to pay for a cryptocurrency) on another competing exchange. An arbitrage event allows a trader to instantaneously sell a cryptocurrency on one exchange for the maximum bid price, buy it on another for the minimum ask price, and make a profit. This trading strategy essentially has no risk and guarantees a profit. It may sound too good to be true, but these events happen.

The problem at hand is that arbitrage events are extremely rare and they don't last long as the exchanges eventually converge in price. Our investigation thus focused on the prediction of arbitrage events. The cause of an arbitrage event can be useful information for a cryptocurrency trader. It allows them to prepare for it and be ready to execute the trade as they can see it coming. This information is also useful for exchanges. The market makers providing the liquidity do not want arbitrage events, as trading is essentially a zero sum game so somebody has lost money when it happens. Ideally their bid and ask prices would be so efficient that arbitrage opportunities never exist.

The Kaiko cryptocurrency data set contains spot price information on 4 markets from 5 exchanges. The markets: Ethereum-US Dollar, Ethereum-US Dollar (Tether), Bitcoin-US Dollar and Bitcoin-US Dollar (Tether). The exchanges: Coinbase, Kraken, Binance, Gemini and Bitstamp. Our investigation focused solely on the Ethereum-US Dollar pair but we expect our results to generalise to the other markets. For each exchange, there is multiple bid prices, ask prices, bid volumes and ask volumes. In advance of our study, we expected sudden changes in prices and liquidity issues to be the potential cause of an arbitrage event. A visualisation of the mid-price data for the Ethereum-US Dollar currency pair, over the 4 exchanges this pair was listed on, is shown in Figure 1.

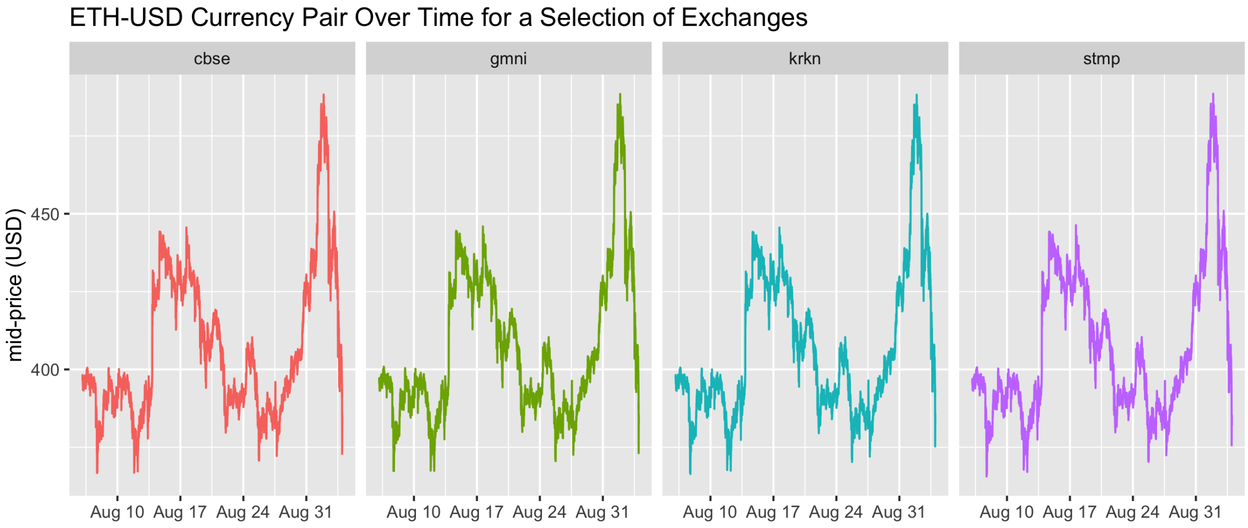

Figure 1: Visualisation of the mid-price associated with ETH-USD across 4 different exchanges, the mid-prices across exchanges do not differ greatly.

Exploratory Data Analysis

Our investigation focused solely on the Ethereum-US Dollar (ETH-USD) market. Unfortunately, there was no information from Binance for this market so we were left with just 4 exchanges. Nevertheless, this gave us minute by minute transactions from 03:19:00 on 06/08/2020 until 23:59:00 on 03/09/2020. This totalled 41,055 observations over approximately 28 days.

Figure 2 is a plot of all the arbitrage events that occurred in the ETH-USD market over the time period. It depicts which exchanges were involved, i.e. where the bid and ask prices were available, the time the arbitrage occurred and the amount it was worth. An interesting thing to note is that the majority of profitable arbitrage events occurred in the final few days of the sample. In total there were 500 arbitrage events, which works out to be just over 17 events per day. To calculate the profit generated by an arbitrage event, we calculated the difference between the maximum bid price and the minimum ask price, and multiplied this by the volume available. To work out the volume available, we took the minimum volume of the bid price volume and ask price volume.

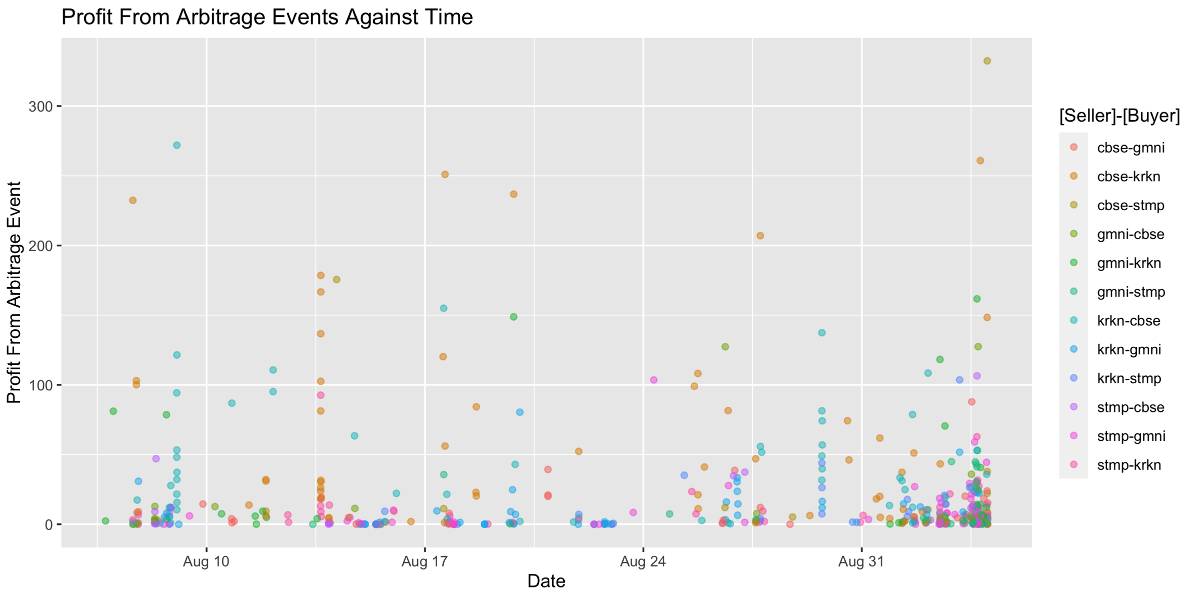

Figure 2: Plot of arbitrage events in the ETH-USD market by exchange pair.

An interesting question posed by Figure 2 is which exchanges are most at fault for the arbitrage events? For example, one exchange may be slower to react to sudden changes in the market prices of the cryptocurrencies. This question inspired Figure 3. It shows the frequency of the arbitrage events by exchange pair. The most arbitrage events happened when Bitstamp has the best ask price and Gemini has the best bid price. This happens 82 times in total. However, when we combine the pairs the most amount of arbitrage events happened between Kraken and Coinbase. In total 129 arbitrage events happened between these exchanges. Kraken leads the way in total number of arbitrage events as it was involved in 274, while Gemini is marginally in second place with 271. These numbers are surprising since it would be assumed that a market like cryptocurrencies, with a large liquidity supply, would be highly efficient.

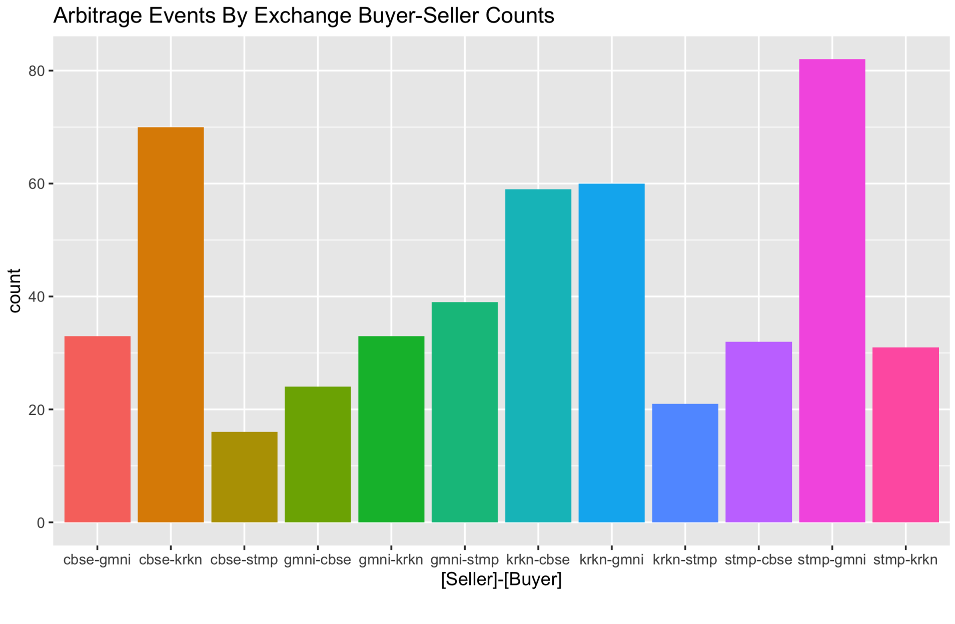

Figure 3: Bar chart of the frequency of arbitrage events in the ETH-USD market by exchange pair.

The distribution of the profits can be seen in Figure 4. Note the y-axis is in the log-scale here since we have many small profits, e.g. nearly 90% of events resulted in a profit of less than $50, and few large profits. The total amount of profit that could have been made by taking advantage of these arbitrage opportunities was $10998.07. Often, in the literature, returns are modelled using a Gaussian distribution or a Log-Normal distribution if they don't cover the entire real line. However, it seems that this data would not fit well to it since there are some outliers in the box plot.

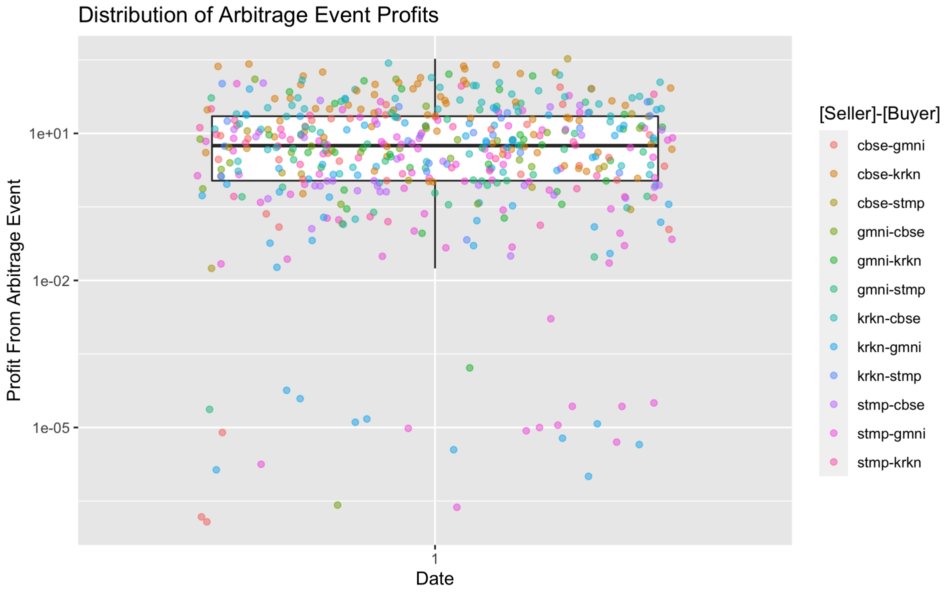

Figure 4: Boxplot showing the spread of log profits from arbitrage events in the ETH-USD market by exchange pair.

Method

The supplied Kaiko dataset, containing spot trades for crypto-currency pairs, was filtered for Ethereum-US Dollar crypto-currency pairs. Investigation of this pair was performed due to its perceived volatility, it was hypothesised that this would lead to more potential arbitrage opportunities, when compared to other crypto-currency pairs. Before analysis could commence the dataset had to be cleaned/wrangled. The steps taken during this data processing steps taken are summarised by Figure 5.

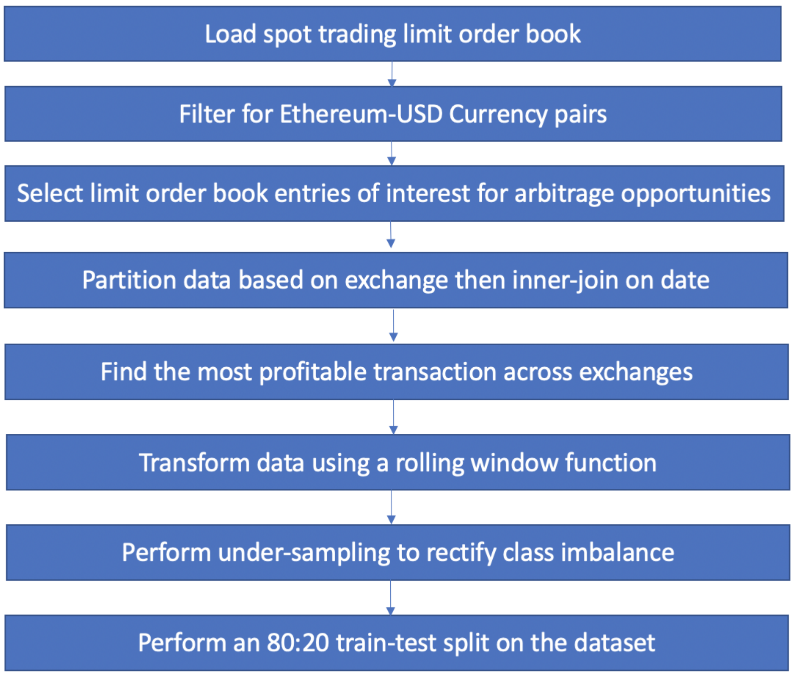
Load spot trading limit order book
Filter for Ethereum-USD Currency pairs
Select limit order book entries of interest for arbitrage opportunities
Partition data based on exchange then inner-join on date
Find the most profitable transaction across exchanges
Transform data using a rolling window function
Perform under-sampling to rectify class imbalance
Perform an 80:20 train-test split on the dataset

Figure 5: Diagram shwoing the steps taken during the data pre-processing phase of this project.

The first step of this process was transforming the dataset containing minutely observations of the limit order book into a more analysis ready form. This was achieved by finding the highest bid price and lowest ask price over each exchange, as these entries in the limit order book would be the most relevant for arbitrage opportunities. The data was then partitioned into sub-datasets according to the exchange that these limit order book entries were occurring on. The split data frame were then inner joined by the date of these limit order book entries of interest, this giving a a wide-format table of 41,055 observations. The maximum profit associated with transferring a currency pair from one exchange to another could now be calculated, this involved calculating the difference in bid to ask price between two different exchanges and multiplying by the minimum volume associated with these two transactions, this would avoid the scenario of the trader holding crypto-currency pair volume that they cannot sell. An arbitrage opportunity was then classified as an observation whose associated maximum trade profit was positive.

The dataset then went through further processing, such that it was ready to be received by a machine learning algorithm. Observations were formed by taking the arbitrage indicator as a response, and the previous 5 timestamps of trading data as covariates, i.e. the bid and ask prices and volumes associated with possible arbitrage opportunities over the different exchanges over the past 5 minutes. Unfortunately, this training dataset had severe class imbalance, with there only being 500 arbitrage opportunities to 40,550 non-arbitrage opportunities. Machine learning algorithms tend to struggle on such problems, often learning to blindly classify observations as the modal class. An Under-sampling approach was implemented to counteract this, whereby the data-set was augmented to include all positive examples and an equal number of randomly sampled negative examples. This approach was deemed appropriate as the under-sampled data-set was still sufficiently large for machine learning to be performed, with a total number of observations equal to 1000, had this not been the case then oversampling would have been considered.

With the data now fully processed it was ready to be used by machine learning algorithms. The data was randomly partitioned into a training and testing set, with a split ratio of 80:20. Several classification algorithms were trained: Logistic Regression, Quadratic discriminant Analysis, Random Forrest and Support Vector Machine (RBF kernel). All of these models were evaluated using 10-fold cross validation on the training dataset, the random forest and support vector machine models had hyperparameter tuning performed on them by using the generated cross validation accuracy estimates. Model selection was then performed by selecting for the model with the highest cross validation accuracy, this was found to be the random forest classifier. Selecting a model this way will bias the aforementioned cross validation estimate associated with the model. A new estimate of the out-of-sample error for the selected model was formed by testing it on the test dataset.

Findings

Training Data Results:

When using a balanced data set and a binary classifier, it is sensible to try and maximise the accuracy of the classifier. Table 1 shows 10-fold cross validation estimates of accuracy and accuracy standard deviations for a selection of models trained to predict arbitrage events. The Random Forest model was the most accurate, predicting 76.26% of the events correctly, the standard deviation of the validation accuracies forming this cross-validation estimate for the accuracy was 5.09%. The Support Vector Machine was the next best predictor with 74.5% predicted correctly. Quadratic Discriminant Analysis produced the lowest standard deviation, but also the lowest accuracy.

Method	Accuracy	Accuracy SD
Logistic Regression	0.6900	0.0642
Quadratic Discriminant Analysis	0.6486	**0.0372**
Random Forest	**0.7626**	0.0509
Support Vector Machine	0.7450	0.0654

Table 1: Accuracy and accuracy standard deviations for the models used to predict arbitrage events.

Table 2 shows the confusion matrices of all the models trained, the cells show the percentage of observations falling within an actual-predicted label regime averaged over the 10 folds of cross-validation sets used. The accuracy is quickly found by summing the percentage of True Positives and True Negatives.

Logistic	0	1
0	37.2	18.0
1	13.0	31.8

QDA	0	1
0	38.8	23.6
1	11.5	26.1

RF	0	1
0	39.6	13.1
1	10.6	36.6

SVM	0	1
0	36.2	11.5
1	14.0	38.2

Table 2: Confusion matrices for the models used to predict arbitrage events. The rows represent the predicted labels and the columns represent the actual labels. The cell values show the percentage of data following into each category averaged over the cross-validation folds.

The confusion matrices displayed in Table 2 can also be used to calculate the sensitivity and specificity, these calculations are summarised in Table 3. If the trading strategy resulting from this model is to be risk adverse, then it is sensible to maximise specificity. This is especially true in this case due to the severe class imbalance in favour of negative observations for the target population, Bayes theorem dictates that a poor specificity will lead to a low probability of correct positive predictions when this model is extended to the target population. The random forest classifier performed best in this regard, achieving the highest specificity (0.7888) across the fitted models. Additionally the random forest classifier achieved the second highest sensitivity (0.7364).

Method	Specificity	Sensitivity
Logistic Regression	0.7410	0.6386
Quadratic Discriminant Analysis	0.7714	0.5252
Random Forest	**0.7888**	0.7364
Support Vector Machine	0.7211	**0.7686**

Table 3: Sensitivity and specificity for the models trained.

Test Data Results:

The best performing model according to the 10-fold cross validation accuracy estimates was the random forest classifier, it was also seen that this model maximises specificity while also maintaining a high sensitivity. The cross-validation procedure also revealed the optimum minimum node size for the random forest, Figure 6 shows this hyperparameter tuning process, the optimum minimum node size was found to be 4.

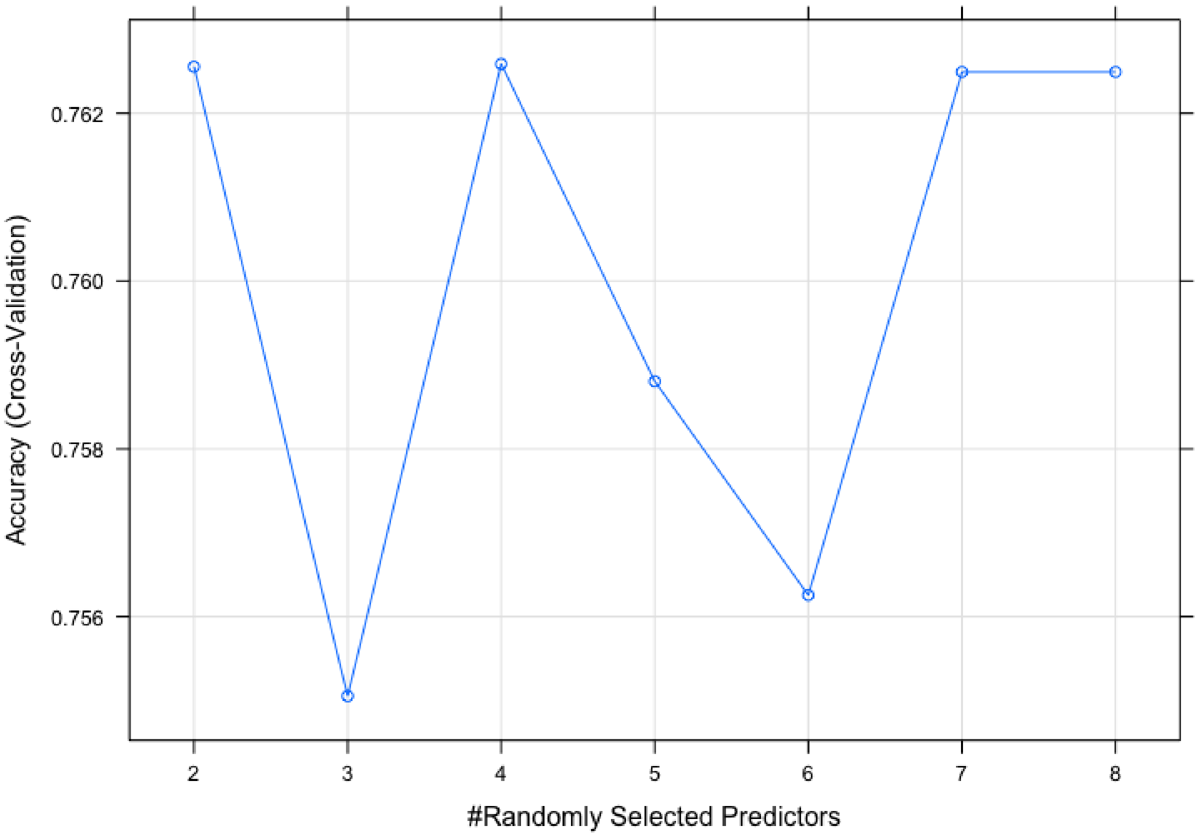

Figure 6: Plot of 10-fold cross validation accuracy across the tuning grid of hyperparameters, this corresponding to the random forest's minimum node size hyperparameter being varied from 2 to 8, the optimum value was found to be 4.

This final random forest classifier with tuned hyperparameters was trained on the full training data set and then tested on the test dataset, the final classification accuracy was 79.50%.

Conclusion

From our analysis, we conclude that there is ample opportunity to find arbitrage events in cryptocurrency markets. Profit can be made by traders if they manage to catch these events. Unfortunately this is difficult to do in practice since there may be only 17 events in a given day that could happen at any point. Our random forest model was able to successfully predict when an arbitrage event would occur, approximately one minute before it would happen. This is a promising result. It shows that the previous few minutes price data and volume can highlight when sudden changes in price or lack of liquidity will cause arbitrage events to occur.

It is impressive that such performance was achieved using relatively simple models, that took low frequency minutely observations over the previous 5 minutes as inputs. There is much scope for further work and improvement relating to this project, higher performance could be achieved given higher frequency data and access to more computational resources. Given additional time, it would be worth while investigating the efficacy deep learning applied to this problem. Recurrent neural networks would be a well suited model for this type of problem, these networks are especially suited for time-series datasets. This approach was not used for this project due to the relatively small size of the under-sampled data set, this approach would certainly be worthwhile investigating given more positive observations.

Given the further improvements previously mentioned, it is envisaged that an algorithm could alert a trader to closely monitor certain situations and increase their profitability. Alternatively, such a model could be used by the exchanges to make their prices more efficient and eliminate ‘free lunches’.