Estimating forward prices in Cryptocurrencies using Machine Learning

Peter Vertenten
CS Graduate Student CCNY
New York, USA
peter.vertenten@gmail.com

Victor Treaba
CS Graduate Student NYU
New York, USA
vtreaba@gmail.com

John Reynolds
CS Graduate Student NYU
New York, USA
jr4716@nyu.edu

Abstract—

Using data from the cryptocurrency exchange Kraken, existing models on moving averages, and machine learning techniques, estimates of forward prices will be generated and tested for their efficacy against smoothed forward prices.

Keywords: cryptocurrencies, Kraken, machine learning, moving averages.

I. Introduction

We have built an application that takes sourced crypto trade data, extracts selected features (aka indicators) and produces a machine learning framework to estimate the change in forward prices of cryptocurrencies. For each cryptocurrency, the features are the cartesian product of the set of all cryptocurrencies estimated against the indicators.

The application allows the modeler to choose various supervised and unsupervised learning techniques, forward time horizons, and features to estimate training parameters that can then be evaluated based on accuracy against our sample of observations.

These models are trained from the superset of cryptocurrencies joined together. We have taken the individual cryptocurrency data and joined them together over 1 minute time windows in order to create features for training the machine learning algorithms for currency estimation. We expected there to be relationships between the various currencies that will impact the machine learning models.

II. Motivation

Cryptocurrency trading markets are relatively new, largely unregulated and yet not saturated with institutional trading. The markets are also active globally and in operation around the clock. This makes cryptocurrency trading unique, volatile and a great place to apply some simple experiments for price prediction. The idea being, given that these are nacisent markets, can simple indicators of years past, be married to machine learning and big data to produce forecasts that will add value in making intraday trading decisions?

Numerous asset price forecasting and trading model efficacy studies are done on daily time series data, where several years of data are required for the study and model periods are typically in the range of several weeks to a hundred or more days.

Crypto markets are one of the few (if only) markets where its possible to collect large volumes of intraday trading data for free and exchanges like Kraken provide an historical API to collect intraday days going back to the origins of the exchange. This provides researchers with a unique opportunity to examine trading models in a high frequency setting, without incurring onerous costs of obtaining the data and linking to various exchanges.

The advent of big data systems (Spark and Hadoop), along with their machine learning libraries(ML) and free historical intraday data, allows us to conduct experiments on large crypto trading data sets to determine if we can get an edge out of simple price prediction models. In addition we can also examine the role of Bitcoin in predicting forward prices of other cryptocurrencies and determine the efficacy.

Cryptocurrency, Big data and machine learning make it an exciting time to apply various models to high frequency crypto data. This paper scratches the surface of what can be done in this space.

III. Related Work

From the origin of trading based on on technical indicators, there has been controversy as to whether its possible to trade profitably by using them. Some of the more common indicators are based on different smoothing techniques of which various types of moving averages are some of the most common to smooth, filter and estimate financial time series data. Early research by LeBaron[13] examined the role of simple moving averages to create buy and sell signals. The signal would generate a Buy if the current price is greater than the moving average and sell signal if its less than. LeBaron's used Dow Intraday data and examined the strategies over 150 day windows. LeBaron's conclusion was that estimating conditional means based on moving averages to generate a trading signal can be extremely dangerous and there was more stability in estimating conditional variances.

As more types of moving averages started to appear, Radys et. al [6] examined the effectiveness of over 10 different types of movings averages and scored their effectiveness based on their ability to smooth vs ability to not severely lag trend reversal. For example as short ranged moving average will do extremely well in not lagging a trend reversal but will not be very useful at smoothing the data. While a longer based moving average will do extremely well at filtering noise, it will have an extremely long lag before it catches up with the trend reversal. Radays were trying to find the moving average that scored relatively well on both. After examining various moving averages they concluded some are better than others and in particular Exponential Hull Moving Average and Triple Exponential Moving Average performed better than other types of moving averages.

Phooi et al [8] examine the use of the various technical indicators which incorporate moving averages into their definition and examine the difference in returns between using the moving averages to make buy and sell decisions as compared to buying and holding. Of particular interest to us is their examination of MACD, Moving Average Convergence Divergence which they credit to Appel [15].

$$MACD_t = EMA_{st} - EMA_{lt}$$

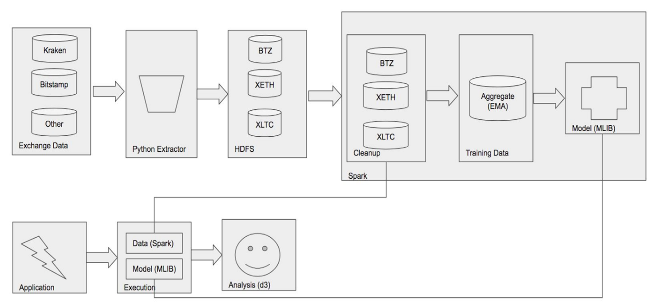
MACD is the difference between a short term moving average signal and a longer term moving average signal. In our experiments we will use a much simplified variant of this as a feature.

The work in financial literature has progressed from examining simple moving averages, to creating more advanced definitions, to looking at differences to generate signals to evaluate efficacy of trading strategies. With the advent of machine learning, researchers are now incorporating signals based on this early work into learning algorithms.

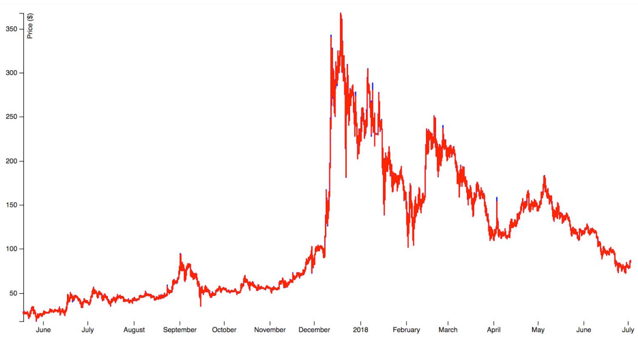
One such study that incorporates early signals with machine learning is Shah [12]. Shah uses machine learning techniques in stock market prediction. In his work, Shah builds a framework using various indicators Exponential Moving Average (EMA), Moving Average, Rate of Change and sentiment analysis with the motivation to see if the models can accurately learn the underlying patterns in stock market time series. Shah's work uses a hybrid approach of incorporating technical models and sentiment analysis into the learning algorithms. Some interesting conclusions in his paper are (1) choice of indicators can dramatically improve/reduce the accuracy of the prediction system, (2) standard linear regression obtained the lowest mse while predicting the EMA pattern and (3) Support Vector Machine combined with Boosting was the only one to give satisfactory results. Shah applied his analysis to daily Open, High, Low, Close and Volume data.

For our Experiments we want to examine the ideas and concepts presented in some of the previously mentioned papers to intraday data. In particular we want to take a variant of moving average difference signals and incorporate them into a learning framework with high frequency intraday cryptocurrency data. Our experiments will differ in that instead of using daily data, we will be using data generated on each trade that occurs during the day.

IV. Application Design



The above diagram shows the high level design of the system. The workflow the data goes through before a machine learning model is created and finally used by the application.



Using these machine learning models we produced the above charts plotting actual price vs predicted price for various cryptocurrencies. As well as calculating the MSE (Means Square Error) of the various models.

V. Datasets

Data for this paper was gathered from Kraken's historical trade API. The historical trade API provides trade price, quantity, time , buyer or seller initiated, going back to 2013. For the experiments we choose four coins with the largest daily trade volume on Kraken which are Bitcoin (BTC), Ethereum (ETC), Litecoin (LTC) and XRP (XRP). The raw data schema is as follows:

Name	**Type**
price	DoubleType
quantity	DoubleType
timestamp	DoubleType
buy_or_sell	StringType
limit_or_market	StringType
_c5	StringType

The raw Bitcoin data set comprised 10.1 million rows, Ethereum 8.1 million rows, Litecoin 2.1 million rows and XRP 3.1 million rows. The first observations for Bitcoin were recorded on 10/6/2013, Litecoin first observations were on 10/24/2013, Ethereum on 08/07/2015 and the last dataset to come online in Kraken, XRP had its first observations on 05/18/2017. In raw sizes, Bitcoin is approximately 452 MB, Litecoin 90 MB, Ethereum 343 MB and XRP 139MB. In aggregate the four raw data sets were just over 1GB.

Its interesting to see the price evolution and volatility of cryptocurrencies prices in this dataset. Bitcoins genesis block occurred on January 3rd, 2009 and the value of Bitcoin was essentially $0.01. By the time Kraken came online (October 6th 2013), Bitcoin's value had increased to $122, which is a 1219900% increase in price in less than a five year period. For Krakens history Bitcoins price ranged from a low of 122 to a high of $19,960!.

Since the study is analysis of the impacts of lagged prices on the set of coins, the datasets were merged and the starting point for the analysis begins on when the last coin XRP arrived to the data set at 8/7/2017 and the last price observations for the study occurred on 7/4/2018. For the merged datasets Bitcoin's price ranges from $1809 to $19960.

While the quality of the data from Kraken is pretty high there are places in the data where exchange outages could have occurred impacting all or some of the coins. There are also intervals where activity occurs in just a subset of coins. To account for some of this the data was aggregated into one minute intervals and any missing minute intervals were filled in with the last price.

For this study we did not use the Kraken supplied data fields, buyer or seller initiated, or wether the trade was one from a market or limit order and field "_c5" . These three columns were ignored. The last column "_c5" was alway empty. We only made limited use of the quantity field for the purposes of interval aggregation and did not incorporate any trade volume changes as indicators of price movement.

The cleaned and processed data aggregated into one minute intervals used in the study has 579K rows and is approximately 397 MB in size. The final data set consists for each currency 1 column for the aggregated price during that interval, 1 column for the lagged moving average price change and 1 column for the forward moving average price change, an id statistic for the column row. The merged schema is below:

Name	**Type**
bucket	IntegerType
partition	IntegerType
XXBTZUSD_price	DoubleType
XXBTZUSD_dpx	DoubleType
XXBTZUSD_fwdDelta	DoubleType

XLTCZUSD_price	DoubleType
XLTCZUSD_dpx	DoubleType
XLTCZUSD_fwdDelta	DoubleType
XXRPZUSD_price	DoubleType
XXRPZUSD_dpx	DoubleType
XXRPZUSD_fwdDelta	DoubleType
XETHZUSD_price	DoubleType
XETHZUSD_dpx	DoubleType
XETHZUSD_fwdDelta	DoubleType

Remediation

The objective is to build an application that can take in trades data from various cryptocurrency exchanges, normalize the data and apply forward price forecast models to the data. The framework developed for the analysis of this paper allows for a header based csv file to be used from any exchange, provided the exchange provides a time, price and quantity for the trades.

The next stage in development of this framework is to provide a trading simulator that will execute buys or sells based on the forecast and some measure of "edge" , where "edge" is a function that takes into account the relative strength of the forecast, overall portfolio position, forecast volatility and price volatility of underlying instruments in the portfolio. It's not recommended to use the forecasts created in the framework for live trading as its unclear without simulation if the forecasts will be "actionable". The forecast to simulator stage will be completely automated, were the results forecast stage are fed into the simulation stage. The stages will look as follows:

Raw Data
⇓
Data cleaning and normalization
⇓
Intermediate Model Results
⇓
Forecast
⇓
Forecast input to simulation
⇓
Simulation Score
⇓
Live Trade Approved Simulations

In the current version a determination can be made if any of the indicators has any efficacy in helping to predict forward prices. For example, in the following sections, there's models that can be easily ruled out because they provide zero explanatory value in forecasting forward price changes. The other observation is the cross currency impact as the models show that some currencies are more useful as features to in the forecasts of coins than others. For example, it's probably a good idea to include some function of the change in Bitcoin prices to forecast the change in Ethereum prices.

Finally, if the models pass the simulation scoring then they could be plugged into a real time trading platform as part of the decision making process of whether to buy or sell a cryptocurrency.

VI. Experiments

We set up a framework to estimate the forward change in prices of various cryptocurrencies using buckets of changes in lagged moving average price. Our experiments are designed to determine the effectiveness of using price differences based on lagged moving average to predict price forward price changes. We are also interested in determining the importance of using Bitcoin price changes as an indicator for predicting other forward price changes. We also look at predicting the forward price changes based on a weighted basket of cryptocurrency.

We setup the following experiments (i) regressing the forward price change of currency C(i) on price change of using a lagged moving average. (ii) regressing the forward price change of currency C(i) on just the lagged price change of Bitcoin. (iii) regressing the forward price change of currency C(i) on itself and Bitcoin's lagged price change (iv) regressing the forward price change of currency C(i) on a basket of coins, using their lagged price change and finally (v) repeating the same experiment in (iv) but removing Bitcoin.

In conducting these experiments we are mindful of a number of issues that arise, such as, that adding more independent variables will improve the insample score and lead to overfitting which tends to happen in price prediction models. To adjust for this the models are evaluated on their out of sample mean squared error (mse). In addition, we also determine the extent of bias or variance in the experiments by running the regressions on the full datasets and comparing the results to randomized training section, to determine the change in mse. In general if the fit score declines with more data it indicates bias in the model and if the fit score declines comparing in-sample to out out of sample it indicates variance ("overfit").

The experiments were run using historical trade prices from the US based crypto exchange Kraken. In working with the data it's important to time align events across all of the coins and then have some measure of the price during that interval. The choice of interval and price summary will have an impact on the results. We expect the length of the interval will have more of an impact than price statistic computed within an interval, but we did not verify this. The data preparation steps used in the model are listed below:

4 Raw datasets
⇓
Condense into 1 minute time intervals
⇓
Fill in missing 1 minute gaps
⇓
Merge based on interval id(aka bucket)
⇓
1 Merged dataset
⇓
Compute lagged/forward moving averages
⇓
Compute lagged/forward price changes
⇓
Run experiments

Data Condensing Step

In the data condensing step, all trades occurring between time $t_1 <= t < (t_1 + 1)$ minute are aggregated into a single line. In a 1 minute interval the number of trades can range from 0 to several hundred. At the high end this will result in a compression of several hundred trades into one line by creating a summary statistic for the price. The summary statistic used for the price on a condensed level is a modeling choice. For this study we choose the weighted volume price (referred to as Weighted Price). Other prices such as average, moving average, ema could all have been just as easily used. The choice of price statistic will change the results, but we think it will be less impactful than the length of the summary interval measured in time. Aggregate Quantity over an interval (AggQty(int)) and Weighted Price over and interval (WgtPrice(int)) are given below:

$$\text{AggQty(int)} = \text{AQ(int)} = \sum_{i=t1}^{t1+1min} quantity(i)$$

$$\text{WgtPrice(int)} = \sum_{i=t1}^{t1+1min} price(i) * quantity(i)/AQ(int)$$

Using this, trades occurring during a one minute interval

Raw Trades

Time	**Price**	**Quantity**
t_0	p_0	q_0
t_1+n_1	p_1	q_1
$t_2+n_1+n_2$	p_2	q_2

⇓

Condensed Row

t_0 rounded to min	$(p_0*q_0+p_1*q_1+p_2*q_2)/(q_0+q_1+q_2)$	$q_0+q_1+q_2$

Data Fill Step

Kraken is a 24 x 7 exchange so there will be minute periods where no activity occurs. When this happens the missing interval is filled in with the price of the preceding period, this is using the premise that the most accurate information is the last price.

Aggregated Price Data with missing observation at t_{0+2}

Time Interval	**bucket**	**WgtPrice**
t_0	24965280	p_0
t_{0+1}	24965281	p_1
t_{0+3}	24965282	p_2

In this example t_0, t_{0+1} and t_{0+3} are observed but not trades occurred during t_{0+2}. Gaps will cause an issue for data merging, so we remediate this by filling in t_{0+2} with t_{0+1}'s $price_1$. Using this assumption the data series will look as follows:

Aggregated Price Data with Fill

Time Interval	**bucket**	**WgtPrice**
t_0	24965280	p_0
t_{0+1}	24965281	p_1
t_{0+2}	24965282	p_1
t_{0+3}	24965283	p_2

Each condensed interval period is given an “bucket” which is a mapping from the time to an integer which will be the same for all time periods for all coins. This “bucket” id will be used in the next phase to merge the 4 datasets into one. It represents minutes since the epoch, the time in which the aggregate trades took place.

Merging Datasets

Once the datasets have been condensed, filled and assigned a bucketid(which will be the same across all products for the same time periods), the datasets can be merged on bucket. From the actual project data sets here is a sample interval for bitcoin (where P_BTC is the weighted price of Bitcoin).

bucket intervals for Bitcoin

bucket	P_BTC
24965280	2586.1
24965281	2585.0
24965282	2589.1
24965283	2588.0

for Ethereum :

bucket	P_ETH
24965280	356.45
24965281	356.42
24965282	356.29
24965283	357.43

the datasets are merged on bucket and to ensure performance when working in Spark the data must be “partitioned” otherwise the analytics can hang during processing. The merged dataset looks as follows :

Merged dataset based on bucket

bucket	partition	P_BTC	P_ETH
24965280	5779	2586.1	356.45
24965281	5779	2585.0	356.42
24965282	5779	2589.1	356.29
24965283	5779	2588.0	357.43

Computing the Lagged and Forward Moving Averages

For the experiments we are going to use the lagged and forward moving averages of the weighed prices over 10 minute intervals. The time horizon choice should have an impact on the price. For these experiments we are determining if the lagged price difference is of any help in forecasting the forward price difference, so we are limiting the window to 10 minutes and using just a single window length (as opposed to fitting with windows of 5, 10, 15, 20 minutes etc).

$$\text{Lagged Moving Avg(int)} = \sum_{i=t}^{t-10} Price(i)/10$$

$$\text{Forward Moving Avg(int)} = \sum_{i=t+1}^{t+11} Price(i)/10$$

As a side note we accomplished this in Spark using “windowing” operations on Dataframes so the code will look as follows:

```
val lagBack =
Window.partitionBy("partition")
.orderBy("bucket")
.rowsBetween(-10,0)

var ma =
sorted.withColumn(
"ma_back", avg(sorted("price")).over(lagBack))
```

Response and Features

For the experiment regressions the response is going to be the difference between the forward moving average and the current price, which will be some measure of actual forward change and the features will be the difference between the current price and lagged moving average:

$$Reponse(i) = Forward\ MA(i) - price(i)$$

the features (independent variables) will be:

$$Features(i) = Price(i) - Lagged\ MA(i)$$

For discussion in paper:

$$dFwd(i) = Response(i) = Forward\ MA(i) - price(i)$$

and

$$dPx(i) = Price(i) - Lagged\ MA(i)$$

Experiment Equations
With the definitions of responses (aka dependent variables) and feature (aka independent variables) the experiments can now be formulated:

Experiment 1 (Self)
Experiment 1 will be regressing the *dFwd(i)* on *dPx(i)*, for each coin. This experiment will assess the effectiveness of using the lagged price difference to predict the forward change in price.

$$dFwdBTC = \beta_1 dPxBTC + \varepsilon$$

$$dFwdETH = \beta_1 dPxETH + \varepsilon$$

$$dFwdLTC = \beta_1 dPxLTC + \varepsilon$$

$$dFwdXRP = \beta_1 dPxXRP + \varepsilon$$

Experiment 2 (JustBTC)
Because of the importance of Bitcoin in the cryptocurrency space, we want to assess the effectiveness of just using Bitcoin as a predictor for forward prices of other coins (Bitcoin on itself was done in Experiment 1)

$$dFwdETH = \beta_1 dPxBTC + \varepsilon$$

$$dFwdLTC = \beta_1 dPxBTC + \varepsilon$$

$$dFwdXRP = \beta_1 dPxBTC + \varepsilon$$

Experiment 3 (SelfBTC)
Experiment 3 examines the effectiveness of regressing *dFwd(i)* on the features (*dPx(i)* for each coin and Bitcoin) so the predictors will be as :

$$dFwdETH = \beta_1 dPxBTC + \beta_2 dPxETH + \varepsilon$$
$$dFwdLTC = \beta_1 dPxBTC + \beta_2 dPxLTC + \varepsilon$$

$$dFwdXRP = \beta_1 dPxBTC + \beta_2 dPxXRP + \varepsilon$$

Experiment 4 (ALL)
This experiment examines the impact of regressing dFwd(i) on the entire set of dPx(i) for each i, so the equations will look as follows:

$$dFwd = \beta_1 dBTC + \beta_2 dLTC + \beta_3 dLTC + \beta_4 dLTC + \varepsilon$$

Experiment 5 (AllExBTC)
This experiments looks to examine the difference in results when Bitcoin is not used as a feature.

$$dFwd = \beta_1 dLTC + \beta_2 dLTC + \beta_3 dLTC + \varepsilon$$

Comments on Assumptions
This basic structure allows us to change a number of factors including :

- The window interval
- The price statistic in the window
- The lag periods of both the forward and reverse ma's
- Using different combinations of lagged price changes
- The fit algorithms, regression, classification, etc.
- Additional features

Results
For each currency we list the outcomes of the experiment ranked by lowest out of sample mean squared error. The results were generated using randomized training and test data sets from the original dataset. The training portion was 75% of the dataset. The ml algorithm was standard linear regression without regularization. For all the runs in total size of the dataset is 579K rows, the training portion is 75% of the dataset and the test 25% (randomized with a fixed seed) .

Ethereum Forward Price Results

Experiment	out mse	in mse	r-squared
All	4.3335033	4.2578881	0.027592
SelfBTC	4.3476822	4.2798235	0.022583
JustBTC	4.3551213	4.3099548	0.015701
AllExBTC	4.3836192	4.3253009	0.012197
Self	4.4208486	4.3786902	0.000004

Not surprisingly regressing just *dFwd* on *dEth* (Experiment 1) did not provided any predictive power. Using all of the coins (Experiment 4) was able to provide some predictive power for the forward price of Ethereum. Bitcoin price change accounted for more than half of the r-squared. Ethereum models were the only models which exhibited higher out of sample mse compared to insample, indicating a slight variance. The out of sample increase in mse was not dramatic.

Litecoin Forward Price Results

Experiment	out mse	in mse	r-squared
ALL	0.4013320	0.4111755	0.086612
SelfBTC	0.4072391	0.4170336	0.073599

AllExBTC	0.4143272	0.4252108	0.055434
JustBTC	0.4219529	0.4337570	0.036450
Self	0.4360162	0.4472443	0.006489

Litecoin Experiment 4 achieved the highest fit score of all the experiments for all of the coins. Litecoins explained price variance was more uniformly captured by all of the coins as opposed to Ethereum where Bitcoin explained more than half the variation in the fit score. As with all of the other coins, Litecoin regressed on itself did not yield a useful predictor (it was slightly better than nothing). Litecoins in sample mse was higher for all experiments indicating bias in the model. Litecoin Experiment 4 could be useful in aiding a trading decision model. One would first have to discount that lower volume of LTC relative to the other coins was artificially reducing its variance and hence making the fit look better.

XRP Forward Price Results

Experiment	**out mse**	**in mse**	**r-squared**
ALL	0.0000229	0.0000246	0.013198
AllExBTC	0.0000229	0.0000247	0.009812
SelfBTC	0.0000229	0.0000246	0.010490
JustBTC	0.0000230	0.0000248	0.005578
Self	0.0000231	0.0000249	0.001465

XRP price variation is not explained well by Bitcoins price variation so its fit score ranking relative to Ethereum and Litecoin is lower. XRP exhibited the highest change between in sample and out of sample mse for all models, indicating a healthy does of bias in the XRP models. XRP regressed on all price changes (Experiment 4) yields some help but it explains less than have the price variation relative to LTC or ETH. It looks like a major contributing factor to this is due to BTC not helping explain much of the forward price change.

Bitcoin Forward Price Results

Experiment	**out mse**	**in mse**	**r-squared**
AllExBTC	683.83892	685.74901	0.003910
ALL	683.92818	685.43642	0.004364
Self	685.77244	688.13481	0.000444

For Bitcoin there are only three experiments, two are redundant (Self and JustBTC) and one is undefined (SelfBTC). Like all of the other coins in the experiments Experiment 1 (Self) was of no help in predicting the forward price. Experiment 5 (AllExBTC) and Experiment 4 (All) were slightly better than using nothing. BTC experiments all exhibited bias as the out of sample mse was lower than the in sample.

VII. Future Work

The work presented in this paper scratches the surface on estimating forward price changes. There are many dimensions to examine, including studying the impact of different types of indicators (for example making use of unused data such as trade initiator), expanding the use of moving average types, studying the decay horizons, adding the impact of other exchanges, measuring the exchange effect, looking at the time of day effect, segmenting the data based on price change regimes (for instance from October of 2017 to December of 2017 cryptocurrencies had gains that went vertical), in addition to applying different learning algorithms to the data.

VIII. Conclusion

We estimated the changes in forward prices of four liquid cryptocurrencies using various moving average price change models. We found that using price changes of all of the coins together yielded the best results. The models were of some help in helping to predict the changes in price of Ethereum and Litecoin. Bitcoin was an important indicator in predicting the price movements of Litecoin and Ethereum (though more important for Ethereum). In general the models exhibited slight bias and unlike a lot of financial price prediction models, they did not suffer from overfit. Our model was most successful in predicting the price of Litecoin and gave strong enough results be to used as a potential trading guide for an execution algorithm (though verification on the smaller trade volume needs to be looked at).

Acknowledgment

We would like to thank the HPC team at NYU as all of the calculations were done on DUMBO. In addition we would like to thank the exchange Kraken as they are one of the few that provides free historical trades data, with an easy to use API.

References

1. T. White. Hadoop: The Definitive Guide. O'Reilly Media Inc., Sebastopol, CA, May 2012.
2. Apache Spark 2.1.0 Documentation, https://spark.apache.org/docs/2.1.0/ml-tuning.html
3. H. Karau, et. al. Learning Spark. O'Reilly Media Inc., Sebastopol, CA, February 2015.
4. S. Ryza, et. al. Advanced Analytics with Spark. O'Reilly Media Inc., Sebastopol, CA, June 2017
5. Apache Spark 2.3.1 Documentation, https://spark.apache.org/docs/latest/ml-classification-regression.html#linear-regression
6. Aistis Raudys, Edmundas Malčius, and Vaidotas Lenčiauskas, Moving Averages for Financial Data Smoothing, https://pdfs.semanticscholar.org/257b/837649d8b50662b3fe2c21fce825a1c184e5.pdf
7. Liu Honggang, Ji Ping, Geng Bixaun, A Comparative study of Intraday Trading Systems Developed based on Moving Average: http://www.worldresearchlibrary.org/up_proc/pdf/154-145354079345-49.pd
8. Jaxinta Chan Phooi M'ng Rozaimah Zainudin, Assessing the Efficacy of Adjustable Moving Averages Using ASEAN-5 Currencies, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5004863/
9. Andrew W. Lo, Harry Mamaysky and Jiang Wang, Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation, http://web.mit.edu/people/wangj/pap/LoMamayskyWang00.pdf
10. Frank Klinker, Exponential Moving Average Versus Moving Exponential Average, http://frank-klinker.de/Paper/Klinker-EMA-MEA-(pre).pdf
11. Luckyson Khaidem, Snehanshu Saha Sudeepa , Roy Dey
Predicting the direction of stock market prices using random forest, https://arxiv.org/pdf/1605.00003.pdf
12. Vatsal H. Shah, Machine Learning Techniques for Stock Prediction, https://shequcdn.bigquant.com/uploads/default/original/1X/5c6d3b9959a8556a533a58e0ac4568dfc63d6ff4.pdf
13. Blake LeBaron, (1999), The stability of moving average technical trading rules on the Dow Jones Index, Brandeis University. , http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.419.6910&rep=rep1&type=pdf
14. Appel G (2005) Technical Analysis: Power Tools for Active Investors. Financial Times Prentice Hall. Upper Saddle River, NJ.