Automated Bitcoin Trading via Machine Learning Algorithms

Isaac Madan
Department of Computer Science
Stanford University
Stanford, CA 94305
imadan@stanford.edu

Shaurya Saluja
Department of Computer Science
Stanford University
Stanford, CA 94305
shaurya@stanford.edu

Aojia Zhao
Department of Computer Science
Stanford University
Stanford, CA 94305
aojia93@stanford.edu

Abstract

In this project, we attempt to apply machine-learning algorithms to predict Bitcoin price. For the first phase of our investigation, we aimed to understand and better identify daily trends in the Bitcoin market while gaining insight into optimal features surrounding Bitcoin price. Our data set consists of over 25 features relating to the Bitcoin price and payment network over the course of five years, recorded daily. Using this information we were able to predict the sign of the daily price change with an accuracy of 98.7%. For the second phase of our investigation, we focused on the Bitcoin price data alone and leveraged data at 10-minute and 10-second interval timepoints, as we saw an opportunity to evaluate price predictions at varying levels of granularity and noisiness. By predicting the sign of the future change in price, we are modeling the price prediction problem as a binomial classification task, experimenting with a custom algorithm that leverages both random forests and generalized linear models. These results had 50-55% accuracy in predicting the sign of future price change using 10 minute time intervals.

1 Introduction

1.1 Bitcoin

Bitcoin is a digital cryptocurrency and payment system that is entirely decentralized, meaning it is based on peer-to-peer transactions with no bureaucratic oversight. Transactions and liquidity within the network are instead based on cryptography. The system first emerged formally in 2009 and is currently a thriving open-source community and payment network. Based on the uniqueness of Bitcoin's payment protocol and its growing adoption, the Bitcoin ecosystem is gaining lots of attention from businesses, consumers, and investors alike. Namely, for the ecosystem to thrive, we need to replicate financial services and products that currently exist in our traditional, fiat currency world and make them available and custom-tailored to Bitcoin, as well as other emerging cryptocurrencies.

1.2 Price Prediction

The Bitcoin market's financial analog is, of course, a stock market. To maximize financial reward, the field of stock market prediction has grown over the past decades, and has more recently exploded with the advent of high-frequency, low-latency trading hardware coupled with robust machine learning algorithms. Thus, it makes sense that this prediction methodology is replicated in the world of Bitcoin, as the network gains greater liquidity and more people develop an interest in investing profitably in the system. To do so, we feel it is necessary to leverage machine learning technology to predict the price of Bitcoin.

1.3 Prior Work

Given that Bitcoin is still a new technology with highly volatile market price, current price prediction models are few and of limited efficacy in a production environment. Most recently, Shah and Zhang [1] described their application of Bayesian regression to Bitcoin price prediction, which achieved high profitability. Current work, however, does not explore or disclose the relationship between Bitcoin price and other features in the space, such as market capitalization or Bitcoin mining speed. We sought to explore additional features surrounding the Bitcoin network to understand relationships in the problem space, if any, while also exploring multiple machine learning algorithms and prediction methodologies within our research. In this way, our thought is that such an exploration will help us cast a wider net and develop a stronger intuition of the problem space, such that we can apply such learning to achieve higher long-run profitability.

2 Materials & Methods

2.1 Data Collection

We collected two sets of data for our project. The first set is daily data with price and 26 additional features about the Bitcoin network and market, described in Table 1. This was acquired from Blockchain Info [2]. The 24-hour time series minimizes noise concerns from higher granularity measurements and minute volatility, and it serves to determine which features were relevant in predicting Bitcoin price. These features include concepts like the market capitalization of Bitcoin as well as the relationship of Bitcoin transaction volume to USD volume.

Our second set of data consisted of 10-second and 10-minute interval Bitcoin price data. The 10-minute data was pulled from the Coinbase API, a large Bitcoin wallet and exchange service based in San Francisco [3]. We collected 10-second price data by building an automated real-time web scraper that pulled from both the Coinbase API and the OKCoin API over the course of multiple weeks [4]. OKCoin is a service similar to Coinbase, based in Beijing, China. The script runs on an Amazon EC2 instance and is stored in a NoSQL database via Amazon DynamoDB. This real-time data collection mechanism allowed us to collect high-granularity Bitcoin price data and accumulate roughly 120,000 unique price points for use in our modeling step.

2.2 Feature Selection

We considered over 26 independent features relating to Bitcoin trading and the Bitcoin network. Of these 26, we selected 16 to use in our initial algorithm with daily data. These features were selected manually on the basis of our research of their significance to the problem we are trying to solve. We also performed forward and backward stepwise selection to additionally corroborate which features may be most meaningful to our model. Because these results were largely different without clear indication as to why, we chose to leverage our hand-selected features due to our pre-existing intuition about their role/impact within the problem space.

FEATURE	DEFINITION
Average Confirmation Time	Ave. time to accept transaction in block
Block Size	Average block size in MB
Cost per transaction percent	Miners revenue divided by the number of transactions
Difficulty	How difficult it is to find a new block
Estimated Transaction Volume	Total output volume without change from value
Hash Rate	Bitcoin network giga hashes per second
Market Capitalization	Number of Bitcoins in circulation * the market price
Miners Revenue	(number of BTC mined/day * market price) + transaction fees
Number of Orphaned Blocks	Number of blocks mined / day not off blockchain
Number of TXN per block	Average number of transactions per block
Number of TXN	Total number of unique Bitcoin transactions per day
Number of unique addresses	Number of unique Bitcoin addresses used per day
Total Bitcoins	Historical total Number of Bitcoins mined
TXN Fees Total	BTC value of transaction fees miners earn/day
Trade Volume	USD trade volume from the top exchanges
Transaction to trade ratio	Relationship of BTC transaction volume and USD volume

Names and descriptions of the 16 features we chose that relate to the Bitcoin network. We leveraged these features in developing a binary classification algorithm to predict the sign change in Bitcoin price based on daily data points. Our data set consisted of these 16 variables collected daily over the course of the past 5 years, since Bitcoin emerged. We specifically looked at the differential of each of these variables, in order to predict the sign of the price change, as opposed to actual price itself, in order to represent this problem binomially rather than via regression. Our training set comprised of the first 70% of these data, while our test set comprised the remainder.

2.3 Time Series Equations

Due to the large occurrence of micro-variations and perturbations in the price of Bitcoin, we also use 10-second interval data to attempt to gain deeper insight. However, as feature data are unavailable or poorly recorded at such small time intervals, we look to the price curve itself to predict future price change. Specifically, with the idea that future price trends can be inferred directly from a linear combination of existing time series data, we construct three time series data sets for 30, 60, and 120 minutes (180, 360, 720 data points respectively) preceding the current data point at all points in time respectively. This means, to predict the price change at a certain time t, we create the datasets

$$t - 180 \leq S_1 \leq t$$
$$t - 360 \leq S_2 \leq t$$
$$t - 720 \leq S_3 \leq t$$

We then run GLM/Random Forest on each of the three time series data sets separately. This will give us three separate linear models: M_1, M_2, and M_3, corresponding to each of the data sets. From M_1, we can predict the price change at t, denoted ΔP_1. Similarly, we have ΔP_2 for M_2 and ΔP_3 for M_3. These values can then be linearly combined to predict the macro price change defined as

$$\Delta P = W_0 + \sum_{j=1}^{3} W_j \Delta P_j$$

Where W_0 is the intercept term representing initial market value at $t = 0$, and W_j is the weight denoting the influence of S_j. This is a linearization problem which we solve with a GLM, yielding the price change prediction.

In addition to using 10-second interval data, we also use 10-minute interval data to gain a longer term picture of the price trends. The reasoning is that 10-second interval data allows for higher frequency trading with much smaller price changes that rarely pass $1 or $2 USD. Although riskier, in order to implement a novel trading strategy, we must be able to make predictions based on larger price changes to achieve higher profitability. Given that there is also latency in current Bitcoin exchanges between issuing a purchase or sale and the actual completion of the transaction, it makes sense to broaden our window to 10 minutes, as 10 seconds may be too short to capture this entire workflow, making our predictions preemtively stale. Like before, we model three time series S_1, S_2, and S_3, which correspond to 1800 minutes, 3600 minutes, and 7200 minutes (180, 360, 720 data points respectively). The same setup before is used, only the prediction is for 10 minutes in the future instead of 10 seconds.

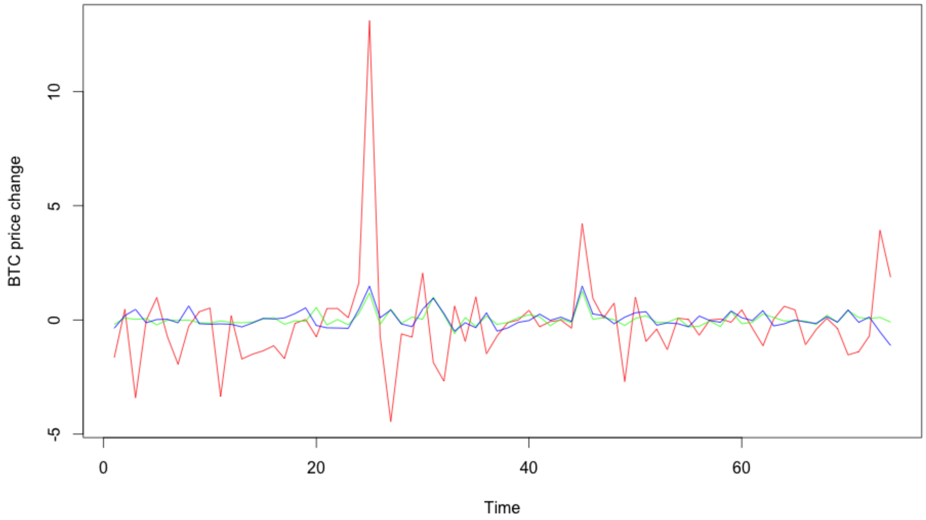
3 Results

STATISTIC	BINOMIAL GLM	SVM	RANDOM FOREST
Sensitivity (TPR)	0.9790	0.0348	1.0
Specificity (TNR)	0.9939	0.5514	0.9392
Precision (PPV)	0.9790	0.0839	0.7762
Accuracy (ACC)	0.9879	0.2716	0.9498

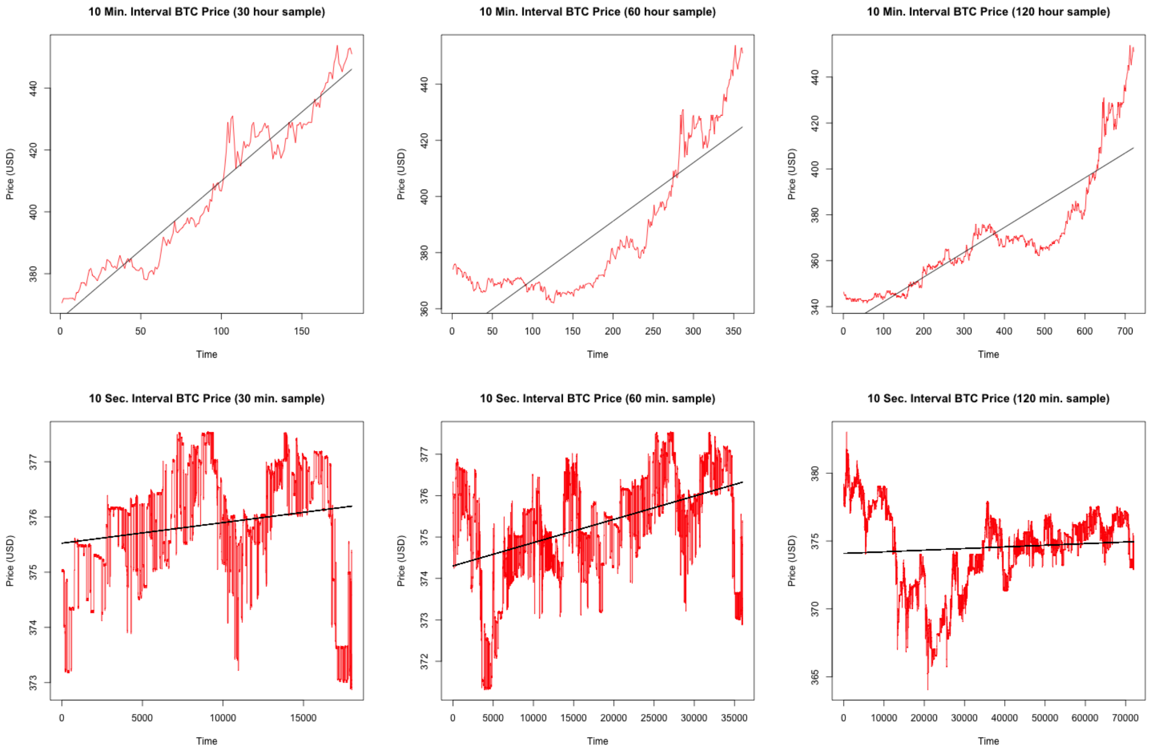
These are results of our performing three different binomial classification algorithms on daily Bitcoin data, using 16 features, described above, to predict Bitcoin price. We leveraged binomial logisitic regression, support vector machine, and random forest algorithms.

STATISTIC	10 SECOND GLM	10 MINUTE GLM	10 MINUTE RANDOM FOREST
Sensitivity (TPR)	0.5429	0.524	0.540
Specificity (TNR)	0.577	0.576	0.619
Precision (PPV)	0.574	0.551	0.581
Accuracy (ACC)	0.085	0.539	0.574

These are results of our performing our prediction algorithm on 10 second Bitcoin price interval data using binomial logistic regression to predict price change. We then switched to 10 minute interval data to diminish the effects of granular price fluctuations, and then ran our algorithm using inner logistic regression and a random forest. We can see that switching from 10 second to 10 minute interval data provided a significant enhancement, while the random forest performed better than the GLM.



These are results from our algorithm's performance on 10 minute interval Bitcoin data. The y-axis represents the change in Bitcoin price, in US dollars, between a Bitcoin price point and its previous Bitcoin price point. The red line is the actual change in price, the green line represents our results from GLM, and the blue line are our results from using a Random Forest. We can see that both of our algorithms perform well, but do not capture the random noise and spikes of variability present in the actual Bitcoin price changes, which is likely the rationale behind most test error.



10 minute interval Bitcoin price with 30 hour sample, 60 hour sample, and 120 hour sample. These figures serve to demonstrate a single example of our linear fits to the 10 minute interval data. Clearly, as our sample size for each linear fit increases, a linear fit worsens in quality though our thought is that 3 linear fits of differing underlying sample

size will allow us to better smooth over noise and random variation that is present in higher granularity data, and in turn, capture and predict the macro actual price change at a given point.

4 Conclusion

4.1 First Phase

The results from the Binomial GLM exceeded our expectations. This can be likely attributed to the long time interval between data points leading to dampening of the price fluctuation within the actual Bitcoin market. Additionally, the higher percentage of true positives compared to true negatives suggests to us that over the longer term Bitcoin prices are generally rising. When applying the SVM algorithm, the observed error rates were much worse than before. This is perhaps due to the need to create artificial separations between data points in higher dimensional space to classify points. As SVM seeks to find underlying patterns within data, perhaps more frequently timed data points are needed to fill in the general trend being observed. As for Random Forest, our results show a high accuracy but lower precision than Binomial GLM. This can be attributed to the fact that Random Forest generates a plethora of decision trees at runtime, and is thus inherently imprecise from run to run, but still yields result that is fairly close to the data. Additionally, the low precision indicates that Random Forest predicts an even larger number of positive trends than data shows, as false positive is the dominating factor.

4.2 Second Phase

With Random Forest and GLM time series models, we find that the 10 minute data gave a better sensitivity and specificity ratio than the 10 second data. Since we are not looking at high frequency trading but predictive trading, the 10 minute data helped show clearer trends. This can be attributed to extremely small and rapid changes in the Bitcoin market that too closely depict the erratic nature of the Bitcoin fluctuations. Additionally, from the graph it is obvious that the prediction is less volatile than the actual data. This is inherently beneficial to financial trading as our strategy can be more lenient with percent changes in buy and sell. Although some short term bursts in price change can be missed out on, the overall gain over a longer timeframe is very promising.

Realistically, in developing a trading strategy, one must know more than simply if the Bitcoin price will rise or drop in the upcoming interval. Specifically, an adroit trading strategy encompasses the degree of change in price and the price that we would find most appropriate to actually make a buy/sell transaction. Thus, it seems possible that 10 minute interval data may be more appropriate in this case, because once we establish a price at which we will buy/sell, the probability of us seeing this price within a 10 minute window is higher than in a 10 second window. As such, with a smaller window, we may predict price change correctly, but have a smaller profitability margin based on the fact that we are unable to dispatch an appropriate trade within the window that our algorithm finds suitable.

We also observed that Random Forests gave us a better accuracy rate as compared to GLM. This is because RFs use nonparametric decision trees, so outliers and linear separability of the data are not concerns.

4.3 Future Work

To improve upon our results in the future, we intend to examine patterns of subsets of the price data. For example, we will break up the data into sequential combinations of 100 data points (100-mers) – these vectors will serve as patterns of how Bitcoin changes over a brief snapshot in time. Then, we can use k-means clustering to cluster these patterns into a number of groups. Then, at new data points, we will pull our most recent 100 data points and use these to represent a new unclassified pattern, which we can then cluster. We will then run our linear regression models on this subset of patterns only. In this way, we are trimming our training set to a set of patterns that is highly similar to our input pattern, which we believe may improve upon our results, at least more so than the current method, which only looks at a price change at a single point in time.

References

[1] Shah, Devavrat, and Kang Zhang. ”Bayesian regression and Bitcoin.” arXiv preprint arXiv:1410.1231 (2014).

[2] Blockchain Info. https://blockchain.info/

[3] Coinbase API. https://www.coinbase.com/docs/api/overview

[4] OKCoin API. https://www.okcoin.com/about/publicApi.do