Predicting Short-Term Cryptocurrency Price Movement with Machine Learning

Group 1: *The 21st Century Gold-Rushers*
C. Jordan Mincey, Lance Simmons, Sinan Ulkuatam, William Young
Department of Computer Science
The University of Virginia
{cjm9fw, lrs4zq, sku3hh, wty5dn}@virginia.edu

Abstract
Cryptocurrencies represent a new, booming market for prospective investors. Since their rise in 2009, both the number and valuation of cryptocurrencies in existence has exploded, resulting in the creation of vast wealth. However unlike traditional financial entities such as securities and bonds, cryptocurrencies are highly volatile and unpredictable. As a result, new methods are needed to assist both personal and institutional investors in making sound investment decisions. In this project, we evaluate various predictive machine learning models toward the end goal of accurately predicting short-term cryptocurrency price movement. Experimental results confirm that this is a difficult problem despite mild success in predicting such movement.

I. INTRODUCTION

Cryptocurrencies are currently booming [1] and are generating large paydays for investors who can accurately predict what the next big cryptocurrency will be. Unfortunately, predicting the value of cryptocurrencies is difficult because they are poorly understood and inherently more volatile than traditional fiat currencies, commodities, stocks, and bonds.

In this work, we utilize machine learning to identify trends driving cryptocurrency price movement with the end goal of predicting the direction of future price movement. At this project's outset, we were unsure how well different machine learning algorithms would handle the highly volatile world of cryptocurrencies and how much the granularity of our data would affect the success of predicting future cryptocurrency prices.

The dataset we used to train our models suffers from a large degree of inherent noise; per-day price fluctuations of 10-15% and above are not uncommon in the world of cryptocurrencies. Such volatility renders difficult any kind of meaningful machine learning on the data since the model must try to find complicated data-dependencies without stability. This challenge is further compounded by the fact that cryptocurrencies have experienced a huge explosion of growth in recent years, skewing some models towards being overly positive.

II. MOTIVATION

Researchers have found that stock market fluctuations can be predicted with some accuracy using technical analysis (TA) [6]. TA employs several naïve techniques to predict stock market trends. For example, TA usually relies heavily on whole-market indices provided by the larger domain of the stock market. In contrast, cryptocurrency markets lack the stability and predictability needed to successfully leverage TA [7]. Cryptocurrency markets are usually void of regulatory bodies, and the currencies themselves possess no actual, physical value. Cryptocurrencies do not have any real backing, and they mainly derive their value from 1) the ability to act as a currency and offer a decentralized medium of exchange that is fungible, portable, non-consumable, secure, divisible, transferable, and immutable; 2) a notion of scarcity in its supply, making the nature of the currency deflationary in general (there are inflationary cryptos as well); and 3) the willingness of people to buy it through cryptocurrency exchanges.

Due to the highly illiquid nature of certain coins, there is the possibility that one may asymptotically approach a zero value without ever completely reaching zero. This makes it harder to define what traditionally one would call a "bankrupt" company, as you can never truly "end" a cryptocurrency and force its valuation to go to zero short of completely compromising its network through persistent 51% attacks. As a result of all these factors, valuation of cryptocurrencies is a different beast when compared to traditional stocks. Whereas traditional stocks have associations with real world factors such as a company's quarterly earnings, product pipeline, or number of employees, the fluctuation of cryptocurrencies is often fueled by hype from zealous investors and fraudsters.

In total, we contend that the valuation of cryptocurrencies poses a fundamentally different problem than that of predicting the traditional stock markets. As such, it stands to reason that experimental machine learning methods may be useful in shedding light on this relatively unexplored problem domain. Furthermore, the first individuals able to successfully identify the key trends driving cryptocurrency prices will undoubtedly amass more wealth than in any fundamental market breakthrough since the industrial revolution.

III. BACKGROUND

Modern cryptocurrencies were born in 2009 with the publishing of Satoshi Nakamoto's paper "Bitcoin: A Peer-to-Peer Electronic Cash System" [2]. The fundamental component of Nakamoto's paper was not the idea of cryptocurrency itself; this had been attempted unsuccessfully in the 1990s. Rather, the distinguishing feature of Bitcoin was the creation of a decentralized system whose ledger could never be altered through the use of blockchains. This ensured that transactions could never be changed after they had been logged. This was the first such system to propose a viable method ensuring that a given transaction was valid without a centralized authority in control.

Cryptocurrencies in their simplest form act as decentralized public ledgers, making transactions publically available yet not trivially digitally unidentifiable. They almost act like a modern-day digital version of gold, where a central authority cannot control its supply or demand while allowing transactions to happen concurrently around the globe. The one problem that has been the bane of cryptocurrencies has been that of scaling, and it is a problem that has been improved on with newer coins that use algorithmic models such as delegated proof of stake, but not on older blockchains such as Bitcoin that use proof of work.

As a result of Nakamoto's landmark paper, many cryptocurrencies were developed. Today, well over eight hundred cryptocurrencies are traded on exchanges, with new tokens added every week. Each cryptocurrency is unique in its exact implementation. Although cryptocurrencies vary vastly in their implementation and intended purposes, they share common ground in the fact that they are all virtual currencies that have some investor-attributed value.

A multitude of exchanges were created that facilitate the trading of cryptocurrencies, usually for fiat currencies such as USD/EUR, but there are also exchanges that let users trade cryptocurrencies for other cryptocurrencies. For example, a user might trade Bitcoin for NEO.

IV. DATASET

Our dataset is comprised of data representing both the stock market and cryptocurrency market. Our stock market data is pulled from Yahoo Finance. Historical cryptocurrency data has been retrieved from a cryptocurrency dataset provided for public use on Kaggle [3]. This dataset includes basic information for fifteen major cryptocurrencies, including highly detailed feature sets for both Bitcoin (BTC) and Ethereum (ETH). While detailed analysis focuses primarily on Bitcoin and Ethereum, owing to the relatively large market size and amount of data related to these popular currencies, we are also working with two popular "alternative" currencies, Ripple (XRP) and NEO. To overcome the small numbers of features available for these two currencies, we have added a number of additional features. These custom-generated features will be described in more detail in Section V.

Composition. The Ripple dataset consists of 1376 daily samples spanning 24 features, with the features representing daily statistics such as open price, close price, and trade volume. The 1376 samples cover the period from December 27, 2013 through October 2, 2017. The Ethereum dataset consists of 789 daily samples spanning 21 features, including similar metrics to the Ripple dataset such as closing price and trade volume. The samples cover the dates between August 8th, 2015 and October 3rd, 2017. The

Bitcoin dataset, our most robust dataset, consists of 2921 daily samples spanning 27 features, representing the period from October 6, 2009 through October 3, 2017. Finally, the NEO dataset consists of 342 daily samples spanning 18 features, covering October 27, 2016 through October 3rd, 2017.

Should our models prove to be able to predict price movement with reasonably high accuracy, we will attempt to implement a trading simulator that places buy and sell market orders based on price predictions. In order to do so, our historical cryptocurrency dataset provided by Kaggle [3] will be augmented with real-time trading information from Bittrex [9]. The Bittrex API will allow us to collect and interact with more recent cryptocurrency market data, including orderbook information and transaction fees. Should we actually implement this functionality in future work, the dataset will be constructed in real-time.

V. PREPROCESSING

We performed light data preprocessing in the form of several Python scripts to correct formatting errors obtained while running the data through scikit-learn. These changes were usually small, such as removing quotes where the models did not want them. We also had data missing in certain datasets where pricing information was available, but the market capitalization was not (NEO). In order to alleviate the issue, we cleaned portions of this data that were either empty or returned null values by simply dropping the offending samples from consideration.

We also generated some of our own attributes for all four cryptocurrencies, based on the attributes that were provided in the Kaggle data set. We added features for three, seven, ten, and twenty day moving averages for both the closing price of the cryptocurrency and the trading volume. We also created an attribute for daily percent change. For the non-Bitcoin cryptocurrencies, we added Bitcoin pricing information as an attribute, since the relative price of Bitcoin often represents a good indicator of the overall cryptocurrency market. To integrate the influence of other financial markets, we included S&P 500 pricing information, extending Friday's closing price over the weekend.

Unfortunately, since NEO does not possess some of the characteristics available to other coins like Bitcoin and Ethereum (including variables such as hash rate, etc.), we had to instead make use of moving averages and volume differentials to serve as predictive attributes. This was successful to a certain extent; in future work we recommend utilizing qualitative variables to predict the price, though such qualitative information is difficult to collect. Finally, since our dataset contains only numeric information, no label encoding or word embeddings were necessary.

VI. EXPERIMENTAL DESIGN

A. Methodology

Classification. Each of the four members of the group selected a cryptocurrency to focus on during the course of the project. Each of us worked on the feature selection and model selection for our selected cryptocurrency. For the first phase of experiments, we exclusively chose to pursue the binary classification problem "*Will the price increase the next day*?" The second phase of our project focused on regression, rather than classification. In general, we adhered to the following workflow:

1. Acquire, parse, and clean the dataset
2. Perform feature selection according to the Recursive Feature Elimination technique
3. Train and test each algorithm in accordance wth double-resampling, using nested 5-fold cross validation to select an optimal model and 5-fold cross validation to test sequestered data using the optimal model
4. Tabulate metrics associate with training and testing results, including confusion matrices, the Matthews Correlation Coefficient, and validation/testing accuracy

Regression. While the above workflow closely describes the methodology for classification, our regression strategy differs slightly in steps 2 and 4. In the regression task, we instead seek to answer the question, "*What will the price be tomorrow?"* When performing regression, we did not perform feature selection. Feature selection is most useful when models are taking a long time to train, or when the dataset is full of spurious or duplicitous information that renders difficult classification, but we observed that our models were training quickly *without* feature selection. Therefore, we opted not to perform feature selection so that we might obtain a bit more accuracy out of our models, as regression is a more difficult task than classification.

B. Algorithms

Bayesian regression has been the most commonly used approach for predicting the price trends of cryptocurrency [4]. In particular, Shah and Zhang's 2014 paper "Bayesian Regression and Bitcoin" started this line of research because they were able to attain high profitability in their study. However, given that Bitcoin's price at the time was constantly increasing, it remains unclear how advantageous machine learning actually proved to be in the generation of the profit. In our own experimental procedure for performing binary classification, we will instead employ a number of non-Bayesian methods. The classification algorithms we have selected to evaluate are:

1. Random Forest Classifier
2. AdaBoost Classifier
3. Decision Tree Classifier
4. Multi-Layer Perceptron (Neural Net)
5. k-Nearest Neighbors Classifier

We believe these models are a good fit for our dataset and problem formulation because we are attempting to predict a binary class for data with many features, rendering clear data dependencies non-obvious. Additionally, these models represent a varied set of approaches to machine learning, and selecting a varied set of algorithms increases the probability that at least one approach provides satisfactory predictive ability. Meanwhile, for regression, we used the following models:

1. Random Forest Regressor
2. AdaBoost Regressor
3. Decision Tree Regressor
4. k-Nearest Neighbors Regressor

Though we had originally planned to utilize the regression-oriented versions of all the algorithms from the classification task, we were unsuccessful in our attempts to properly tune the multi-layer perceptron. Despite multiple attempts to produce "good" results by adjusting the solver and hidden layer sizes, the predictive ability of the MLP was very poor, nearly predicting 0.0 for every sample. With a more advanced neural network architecture it is likely possible to get very good results, since—with enough hidden layers of adequate size—MLPs are capable of approximating any continuous function with minimal error.

C. Model Selection

Classification. For model selection, we performed two nested rounds of 5-fold cross validation in accordance with the double-resampling principle. First, the dataset is split into 5 folds. Within each fold, we again perform 5-fold cross validation, splitting the training data into training and validation sets. For each inner-fold, we train a model on the inner-training data and record validation accuracy along with model parameters. Once all 5 inner folds have been tested against their respective validation sets, we select the model associated with the highest validation accuracy and run the sequestered test data through this model. We repeat this process for each of the five outer-folds, resulting in five test accuracies representing differently-tuned models of the same algorithm. In the results section, we present the average of these five test accuracies as the overall test accuracy obtained for each algorithm.

Regression. We approached model selection in a similar manner to that of classification except that instead of measuring classification accuracy, we measure the R-squared value, which describes how well the data fits the regression model. Intuitively, the model that exhibits the highest R-squared value (maximum 1.0) *best* represents the underlying "function" generating the actual data, which we would like to accurately predict. Following the completion of all five inner folds, we therefore select the model associated with the highest R-squared value and run the sequestered test data through this model. This is repeated for all five outer-folds, resulting in five R-squared values for differently-tuned models of the same algorithm. The overall R-squared for a given model is the average of these five values.

D. Metrics

Classification. In the classification task, our two output classes are *price increases tomorrow* and *price decreases tomorrow*. We treat a 0% change in price as a price decrease, though we are not aware of any instances where no price change occurs. We calculate prediction accuracy and MCC (Matthews Correlation Coefficient) for these classes. For the problem, accuracy represents whether our models are able to correctly predict the next day's price increasing or decreasing. The MCC is a measure specifically applied to binary classification problems that accounts for datasets in which the the number of positive and negative samples are highly skewed. For this metric, a value of +1 corresponds to a perfect classifier. A value of -1 corresponds to a perfectly wrong classifier (a classifier that is wrong every time). A value of 0 in this metric corresponds to the classifier being equally good as assigning classes randomly.

Regression. For the regression problem, we primarily judge the predictive ability of a given model based on its R-squared value, though we also calculate the mean-square error (MSE) as another useful quantitative metric. The R-squared value is a statistical measure of how well the data fits the regression function—in this case, how well the data fits one of our trained machine learning models. In general, the higher the R-squared value (up to 1.0), the better a given model fits the data. Similarly, the MSE measures the average squared distance between the output of a model and the ground-truth value. By squaring the difference, large errors are heavily penalized.

VII. CLASSIFICATION RESULTS

	Bitcoin		*Ethereum*		*Ripple*		*NEO*	
Model	**Accuracy**	**MCC**	**Accuracy**	**MCC**	**Accuracy**	**MCC**	**Accuracy**	**MCC**
Random Forest	59.04%	**0.233**	**56.8%**	**0.14**	**57.9%**	**0.139**	**61.3%**	**0.22**
AdaBoost	59.95%	0.193	54.1%	0.12	56.4%	0.090	51.4%	0.03
Decision Tree	**61.10%**	0.229	49.66%	0.07	53.7%	0.025	55.4%	0.15
MLP	56.58%	0.104	48.64%	0.03	53.5%	0.007	53.2%	0.07
k-NN	58.04%	0.153	45.9%	0.08	53.6%	0.033	58.1%	0.15

Table 1. Model comparison across all four cryptocurrencies during the classification task

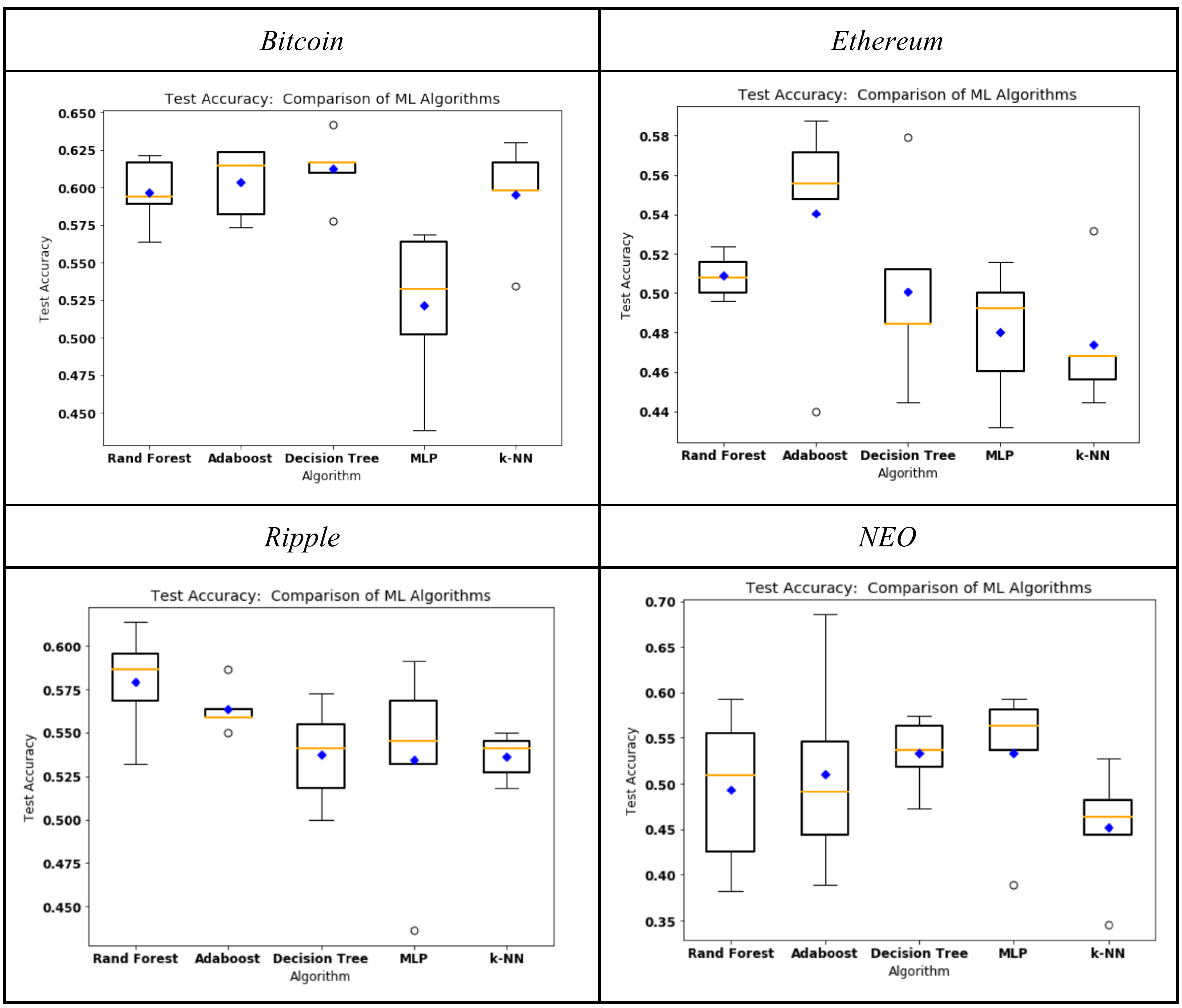

Figure 1. Statistical summary of model testing during classification task.

VIII. REGRESSION RESULTS

	Bitcoin	*Ethereum*	*Ripple*	*NEO*
Model	**R-squared**	**R-squared**	**R-squared**	**R-squared**
Random Forest	**0.9958**	**0.9914**	**0.9881**	**0.9967**
AdaBoost	0.9841	0.9837	0.9783	0.9897
Decision Tree	0.9925	0.9828	0.9717	0.9899
k-NN	0.9948	0.9365	0.9763	0.9953

Table 2. Model comparison across all four cryptocurrencies during the regression task

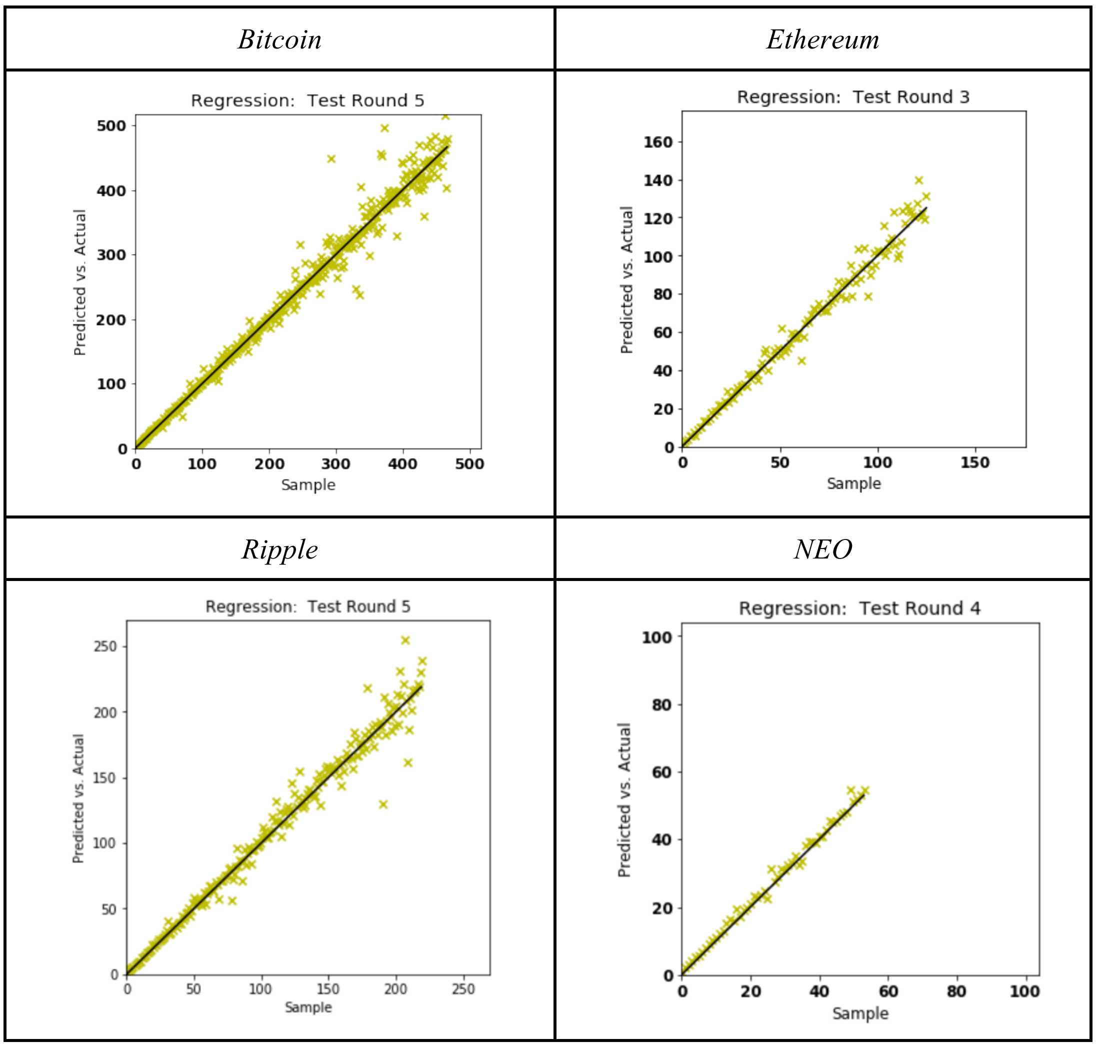

Figure 2. Visualization of regression quality for best Random Forest regressor models. The black line of slope 1.0 represents perfect prediction accuracy, i.e. the prediction matched the ground-truth. Points above or below the black line represent predictions above or below ground-truth, respectively.

IX. DISCUSSION

Model Comparison. For classification, we achieved more-or-less similar results across the five models we selected (Random Forests, AdaBoost, Decision Tree, MLP, kNN) for each given coin. We saw more variance between *coins* than models. For example, NEO had models achieving barely over 50% classification accuracy, whereas Ripple (a relatively stable coin backed by members of the banking industry) achieved accuracy around 55% for all of the models. The largest disparity between model accuracies was observed with Bitcoin. Bitcoin achieved an average of around 60% accuracy for all models except the multilayer perceptron, for which the accuracy was around 53%. This disparity in accuracy is likely due to the MLP not being optimally configured. In general, Random Forests vastly outperformed the other models on the classification task. While we do exceed our goal of predicting short-term price movement with greater than 50% accuracy, the results indicate the existence of additional factors driving cryptocurrency prices not present in our dataset.

For regression, we achieve very high R-squared values, with the best models achieving values exceeding 0.99. It is important to note that such numbers may overstate the value of the regressors. We only need a regressor to make a decent guess to achieve high R-squared values. A regressor that predicted the next day to have the same value as today might deliver a “good” R-squared value, for example, as this would likely not be far off from reality on an average day. If you examine the actual regression plots in the results section, you can see that there are many samples for which the regressor makes poor predictions (likely due to rapid shifts in the coins’ prices), but these samples are outweighed by the much larger quantity of samples for which the regression model does a satisfactory job. Among the regressors, Random Forests performed best on all coins, while Adaboost performed worst on all coins except Ethereum. Random Forests have demonstrated widespread success across a variety of fields, so it’s perhaps no surprise that they would perform well here. In particular, it makes sense that they would outperform Decision Trees, since Random Forests are simply ensembles thereof. AdaBoost’s comparatively poor performance might stem from models overtraining on bad samples and achieving worse performance on the relatively small changes that occur between most days.

Theoretical Application. Our models could theoretically be used to make predictions on the market with real money. Although we have found few academic papers tackling the issue of up-down price direction prediction as applied to cryptocurrencies, existing results such as this Master’s thesis [10] achieved comparable results to ours. Many works in this space can be found scattered across Github repos and personal blogs. Among these works, classification accuracy of ~55% is common.

However, simply achieving a greater than 50% accuracy is insufficient for our solution to make a profit. It is likely that we would need to do a significant amount of extra work to determine when the best times to buy and sell coins are, since the act of trading itself is limited by transaction fees and latency with the coin exchanges. Furthermore, our experimental procedure would need to be modified in order to output a usable predictive model; in this paper, we are less concerned with the specific parameter values of the optimal model than we are with ensuring that models with varying parameters are trained to minimize overtraining. For example, we train and validate decision trees with depths of 3, 6, 9, and 12 in order to ensure that the optimally-selected model within each testing fold is general enough to yield good accuracy when faced with previously unseen data. If we instead wished to design an auto-trading application, we would need to select a *single* optimal model from among the five models produced in this experiment (each of the five test rounds uses a differently-tuned "optimal" model obtained during validation). A simple implementation of this selection process would involve *triple* resampling, wherein a *second* outer fold is added and the entire experimental procedure described in this paper is applied only to the training split of this second outer fold. Then, the best-performing *test* model would be applied to the sequestered test data of the second outer fold.

X. CHALLENGES AND FUTURE WORK

A major challenge we encountered in our project was the high level of volatility in price movement, which made accurate prediction with our dataset quite challenging. Predicting the price of cryptocurrencies is at least as challenging as predicting the stock market and perhaps even more challenging. As far as we know, there exists no "oracle" algorithm for predicting movement in the securities market, therefore we did not expect to easily discover one for the vastly more challenging cryptocurrency market. Thus, we were faced with the reality that obtaining accuracy greater than 60% would be challenging.

It's important to note that we *were* able to predict price movement better than chance; a simple coin toss would achieve 50% accuracy, and we were consistently able to beat that measure. In the case of Bitcoin, however, the price falls for approximately 54.3% of the time, thus if we simply predicted *price will decrease tomorrow* for every sample, we could achieve 54.3% accuracy. While our prediction accuracy does in fact outperform this number, the cost of misclassification is severe in the world of price prediction, and numerous online blogs have evidence of cryptocurrency enthusiasts losing significant amounts of money in the pursuit of an oracle trading algorithm.

We also have run into problems with the level of noise in the price of cryptocurrencies. For example, sometimes the price of the cryptocurrency only moves up or down by 0.01% in a day. The classifier models lack have the ability to differentiate a 10% movement from a 1% movement, yet the difference is highly significant for trading purposes. We developed regression models to compensate for this deficiency, but as previously mentioned, R-squared values can be very high even if we mispredict the binary up/down movement each day, rendering algorithmic trading nearly impossible.

XI. CONCLUDING REMARKS

Cryptocurrency price fluctuations represent a noisy, real-world data source with significant practical applications. Of paramount interest to most cryptocurrency and financial enthusiasts is the task of designing and implementing profitable trading strategies, a significant challenge that requires accurately predicting future returns based on past performance and technical indicators. Bearing this in mind, we selected a number of predictive models to address the classification and regression tasks for a group of commonly traded, high market-capitalization cryptocurrencies. Our classification models achieved comparable results to existing research [10]. Our regression models achieved good fit to the data, but these results may overstate the practical applicability of our models; a small error in the wrong price direction, though unsubstantial and relatively inconsequential to the R-squared value, poses serious problems for designing a profitable trading algorithm since many small losses and many small gains will effectively cancel-out one another.

Ultimately, the cryptocurrency market, much like that of the traditional securities market, is defined just as much by side-channel, qualitative information as it is by purely technical information. If the development team for a given cryptocurrency delivers a good speech or releases a promising infographic, the price may *double* in a single day, only to crash the next day. Such violent fluctuations in price without accompanying technical data make it exceedingly difficult to estimate the next day's price, much less the price next week, month, or year, using price and volume data. As a result, an analysis like ours, while interesting, is unlikely to result in an oracle model. This project served to confirm that belief.

XII. REFERENCES

[1] ArXiv, E. T. (2017, August 28). Bitcoin's dominance of the cryptocurrency market is set to fall. Retrieved November 27, 2017, from https://www.technologyreview.com/s/607947/the-cryptocurrency-market-is-growing-exponentially

[2] Bitcoin - Open source P2P money. (n.d.). Retrieved from https://bitcoin.org/bitcoin.pdf.

[3] S. Rajkumar. Retrieved from https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory

[4] Shah, D., & Zhang, K. (2014, October 06). Bayesian regression and Bitcoin. Retrieved from https://arxiv.org/abs/1410.1231

[5] Yang, S., Qiao, Q., Beling, P., Scherer, W., & Kirilenko, A. (2012, May 04). Gaussian Process Based

Trading Strategy Identification. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2051138

[6] Technical analysis. (2017, November 06). Retrieved from https://en.wikipedia.org/wiki/Technical_analysis

[7] Technical Analysis And Cryptocurrencies - Why You Should Be Wary. (2017, July 14). Retrieved from https://storeofvalue.github.io/posts/technical-analysis-and-cryptocurrencies/

[8] Milosevic, N. (2016, March 02). Equity forecast: Predicting long term stock price movement using machine learning. Retrieved from https://arxiv.org/abs/1603.00751

[9] The Next-Generation Digital Currency Exchange. (n.d.). Retrieved from https://bittrex.com/

[10] McNally, S., Predicting the price of Bitcoin using Machine Learning (2016, August 22) Retrieved from http://trap.ncirl.ie/2496/