Towards a Twitter-Based Prediction Tool for Digital Currency

Mason McCoy[1], and Shahram Rahimi[2]
[1]Department of Computer Science, Southern Illinois University, Carbondale, USA
[2]Department of Computer Science and Engineering, Mississippi State University, Starkville, USA

Abstract— *Digital currencies (cryptocurrencies) are rapidly becoming commonplace in the global market. Trading is performed similarly to the stock market or commodities, but stock market prediction algorithms are not necessarily well-suited for predicting digital currency prices. In this work, we analyzed tweets with both an existing sentiment analysis package and a manually tailored "objective analysis," resulting in one impact value for each analysis per 15-minute period. We then used evolutionary techniques to select the most appropriate training method and the best subset of the generated features to include, as well as other parameters. This resulted in implementation of predictors which yielded much more profit in four-week simulations than simply holding a digital currency for the same time period--the results ranged from 28% to 122% profit. Unlike stock exchanges, which shut down for several hours or days at a time, digital currency prediction and trading seems to be of a more consistent and predictable nature.*

Keywords—Cryptocurrency, Twitter-Based Prediction, Tweet Analysis, Crypto Price Prediction, Currency Price Prediction.

1 Introduction

The global economy is comprised of the exchange of currency and goods. Digital currency, also known as cryptocurrency, is rapidly becoming a major force in the economy, with its value already in the hundreds of billions of US dollars and with traders residing in numerous countries [21]. The trade of digital currency is often executed similarly to the stock market, which is a well-aged mechanism. However, with regulations and often a difficult entry level (such as per-trade fees and the limitation of buying stocks in integer amounts), the major direct actors in the stock market are primarily experienced individuals and computer algorithms [3].

Digital currencies are different--they provide an unregulated system of global trade with very small minimum trades, very high volatility, a young market with many novice investors and few algorithms that are well trained for it [8] [9] [10]. While each stock's value comes from the corporation it supports, digital currencies' value comes only from individuals who trade with it based on its popularity and technological aspects. The digital currency market also never rests, while the stock market is closed for trading more than it is open. While corporate insolvency could cause a stock to lose all its value in an instant, a digital currency could lose all its value only if the network ceased to function or if all the users decided it was no longer worth anything. Despite the differences, it is possible to create a profitable algorithm for trading digital currencies, but all the features and trading strategies must be re-evaluated due to the markets' differences.

Developing an algorithm to predict highly volatile digital currency prices presents an opportunity to yield lucrative profit margins. With perfect prediction of whether the price will increase significantly, decrease significantly, or stay close to the same over the next 15-minute interval for just one digital currency, it would be possible to realize hundreds of percent in profits in a matter of weeks, even as the value of that cryptocurrency decreases.

The primary objective of this study was to use machine learning to predict price fluctuations in the aforementioned manner with sufficient accuracy to achieve profits in the real world. The secondary objective was to experiment and develop features and analysis techniques to optimize the prediction accuracy for each of the target digital currencies (Bitcoin, Ethereum, and Litecoin; also known as BTC, ETH, and LTC).

2 Background

2.1 Digital Currency Trading

Digital currency can be traded for fiat currency (such as United States dollars) much like stocks on centralized exchanges, although it can also be freely traded directly between individuals like any commodity. On centralized exchanges, the price depends on the presence of "maker" orders--that is, purchase or sale orders that are not filled immediately and therefore add liquidity to the market. On Coinbase Pro (previously called GDAX) [1], our exchange of choice, maker orders do not incur any fee. However, taker orders incur a fee of 0.25% (for exchanging Bitcoin and USD) or 0.3% (in other cases).

Fig. 1. Coinbase Pro (AKA GDAX) Maker Orders

We have included Figure 1 from Coinbase Pro to give an example of what market price slack looks like on an average day. The green represents maker orders for purchasing Ethereum, while the orange represents maker orders for selling Ethereum. The point where the different colors meet is the market price,

and the height of the line represents the number of Ethereum being requested/offered in the orders between the market price and the price shown on the horizontal axis. The image shows that selling $142,841.59 worth of Ethereum would cause the price to drop to $521.58 (-0.18%). Dividing $142,841.59 USD by 273.50646973 Ethereum shows that the average price of the sold Ethereum would be $522.26, not the displayed market price. The price slack on a purchase for this amount made at this time would therefore cost the seller 0.054% on top of the 0.3% taker order transaction fee.

2.2 State of the Art

Balaji, Paul, and Saravanan [4] surveyed stock market prediction methods and found that a technique using mood analysis of tweets gave very good results even without considering other possible predictors. Desokey, Badr, and Hegazy [5] used k-means clustering to obtain satisfactory results in predicting stock prices. Kamble [14] utilized several different market indicators and decision trees to predict short-term stock market trends. Mao, Zhang, and Fan [15] used a genetic algorithm to select the features to feed into a support vector machine for stock market prediction. In [16], Luo et al. showed that linear regression outperforms the decision tree and random forest methods in predicting stock prices.

Few published papers apply specifically to digital currencies. Vo and Xu [17] predict Bitcoin prices using support vector machines and neural networks. Shehhi, Oudah, and Aung [18] attempted to identify the factors that give digital currencies their value by performing a survey of only 134 individuals. In [19], Laskowski and Kim performed natural language processing on tweets and internet relay chat (IRC) but only calculated the correlation between those messages and Bitcoin price and trading volume. Fallahi [3] used GDELT, a database of global news, in an attempt to predict both stock and Bitcoin prices. Finally, Phillips and Gorse [20] actually used data from social media to predict Bitcoin, Ethereum, and Monero (another digital currency) price bubbles via a hidden Markov model, showing as much as 98.93% profit over buying and holding a cryptocurrency for the same time period.

None of these publications applied a genetic algorithm to select the best machine learning method or parameters for digital currency price predictions, nor did they generate heuristics similar to ours.

3 Methodology

The methodology is broken into the following components: data collection and filtering, feature generation, genetic algorithm, learning, and scoring. Each of these components is briefly described in the subsections below. Figure 2 illustrates the data flow among these components and some of their major constituent parts.

Data Collection and Filtering. Tweets are collected using the real-time Twitter API, then filtered by language and a quick-and-dirty junk detection process. Trading data is collected from the Coinbase Pro historical candles API. Additionally, historical between-cryptocurrency-and-fiat currency trading volume is obtained from the CryptoCompare API [6] for use in the text analysis.

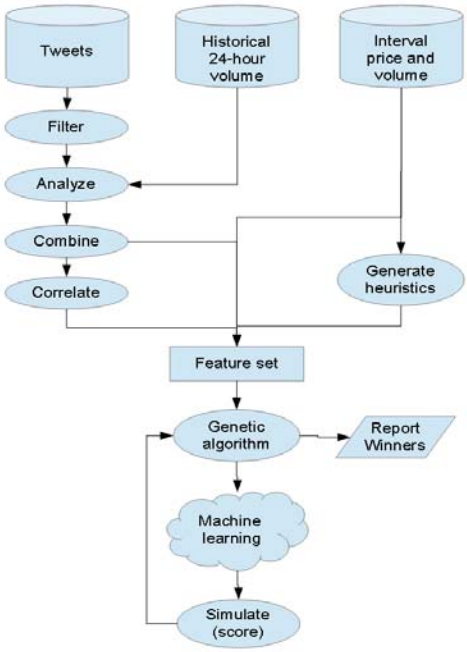

Fig. 2. Data flow Diagram

Feature Generation. The software analyzes tweets in three different ways depending on the language. For English tweets, first an "objective analysis" is performed, followed by sentiment analysis using VaderSharp [2]. For Japanese tweets, only the "objective analysis" is performed, but due to differences between the languages, there is a separate analyzer for Japanese. Once the prediction software has all this data, it generates the features shown in Table I as inputs for the learning algorithms.

TABLE I. FEATURES FOR MACHINE LEARNING

Heuristics	Raw data
Day-of-week price rise probability	Previous period trading volume
Day-of-week price drop probability	24-hour price change
Time-of-day price rise probability	Last 4 periods' price changes
Time-of-day price drop probability	**Analyzed data**
Pay day	Tweet objective impact × correlation
Price rise resistance	Tweet sentiment × correlation
Price drop resistance	
Relative strength index (14, 480, 1344)	

Genetic Algorithm. The software employs a genetic algorithm to turn features on and off, to adjust weights and some parameters of the features, to select and set the machine learning algorithm and its parameters, and to switch the trading strategy. A specified number of individual chromosomes (collections of these settings) are trained per generation, then the better half of that population are cloned, cross-bred, and/or mutated before inclusion in the next generation.

Learning. Machine learning is performed by the package Accord.Net [7] via k-means clustering, a support vector machine using stochastic gradient descent, or linear regression using ordinary least squares, depending on the learning method set in the chromosome. A separate instance is trained for each cryptocurrency for each chromosome because many of the inputs differ by cryptocurrency. All but the last four to twelve

weeks of the available data are used for training, and the remainder is used for scoring.

Scoring. Each chromosome is given one score per cryptocurrency by simply running a trading simulation on the test data (four weeks' worth, from February 9 through March 9). For comparison with simply buying and holding a single cryptocurrency for the same duration, the score is based on how much additional cryptocurrency the algorithm can gain through trading.

3.1 Data Collection and Filtering

Data is obtained and saved by three programs to maximize its availability. The first program, a simple Python script developed by one of our peers, receives tweets via Twitter's real-time feed API [11] and saves them in a Mongo database. The second program, GDAXPrices, collects data from the Coinbase Pro API [1] and makes it available to other programs via a TCP connection. It also saves the data it receives in a CSV file so that historical data is always available. The historical data includes time, starting price, ending price, lowest price, highest price, and volume traded on Coinbase Pro during that 15-minute interval. The third program, NewsChipper, loads old tweets from the Mongo database and receives new tweets in real-time from Twitter. It then filters and analyzes them, combining all the tweets in a 15-minute interval into a single floating point impact for each digital currency and for each type of analysis, resulting in a total of six values per interval. It also obtains between-cryptocurrency-and-fiat currency trading volume information from CryptoCompare [6] for use in the objective analyzer.

TABLE II. FILTER FLAGS

Flag	Behavior
AlwaysIgnore	filter out tweet if >= 1 words have this flag
PossiblySpam	filter out tweet if >= 3 words have this flag
Bitcoin	include for BTC impact
Ethereum	include for ETH impact
Litecoin	include for LTC impact
All	include for BTC, ETH, and LTC impact

NewsChipper's tweet filtering is performed as follows. First, tweets that are neither English nor Japanese are discarded. Next, the tweet is scanned for tokens (words or symbols) that have filter flags specified in the code. The flags are described in Table II. Some words are marked with the AlwaysIgnore flag because we decided the presence of those words likely renders the entire tweet irrelevant, such as "airdrop" (the act of giving away free units of an obscure cryptocurrency in order to gain popularity). Other words are marked as applicable to Bitcoin, Ethereum, Litecoin, or all three. In addition to whole words, a handful of currency symbols (such as '$', '₸', and '€') are marked with the PossiblySpam flag, because we identified many tweets as automated price announcements. If a tweet contains at least one AlwaysIgnore word or at least three PossiblySpam tokens, it is dropped. Also, if a tweet has no digital currency applicability flags, it is dropped. Otherwise, English tweets are given to VaderSharp [2] for sentiment analysis, and both English and Japanese tweets are given to a separate objective analysis engine for each applicable currency.

3.2 Feature Generation

NewsChipper executes sentiment analysis using VaderSharp [2] and objective analysis using our own tool on each tweet. VaderSharp is a port of VADER, which is described as "a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media." [12] Due to its complexity, objective analysis is separately described later in the paper. For each 15-minute interval, a single impact value per cryptocurrency is calculated by summing all the sentiment analysis composite results (which range from -1 to 1); similarly, objective analysis results are also summed into a single impact value per cryptocurrency for each 15-minute interval.

The exact feature given to the machine learning algorithm, rather than directly plugging in the interval's impact sum, is generated by using one of the three different methods of cross-correlation. The first method is generic cross-correlation, which given values from two time series, identifies the time offset at which correlation is the strongest. However, it seems highly unlikely that the market actors (primarily humans) would be equally active at all times, so we developed a modified version of cross-correlation, which we refer to as periodic cross-correlation. Thus, the second and third methods are periodic cross-correlation splitting by 15-minute-period-of-day and 15-minute-period-of-week, respectively. Periodic cross-correlation is performed much like generic cross-correlation, but a different output is given for each period in the specified time frame. For example, period-of-day gives 24 * 4 = 96 separate cross-correlation results no matter how many inputs are given, while period-of-week gives 24 * 4 * 7 = 672. We further extended each cross-correlation method to output multiple time offsets (with one correlation coefficient each) instead of only the time offset with the greatest cumulative correlation coefficient. The number of offsets and correlation coefficients to consider is one parameter in the chromosome and ranges from zero to ten. Because of the lack of space, implementations of these cross-correlation methods are not included in this paper and will be published in an upcoming journal article.

Three pairs of heuristic features are also generated and (depending on the chromosome) included as inputs for the machine learning algorithm. These are naïve probability of rising (and dropping) on that day of the week, naïve probability of rising (and dropping) for that period of the day, and price rise (and drop) resistance, which is an inverse approximation of the amount of price slack based on recent price fluctuations. If more historical data were available, the price change resistance features could be better estimated, but we had to design our own formula for this estimate. This formula requires two inputs. The first is the number of periods (capped at 15) since the price last differed from its current price in the desired direction. In other words, calculating price rise resistance requires determining when the price was last higher than it currently is, while checking price drop resistance requires determining how long it has been since the price was lower. The second is the difference in price at that period versus the current time, represented as a positive fraction (.01% is assumed when the number of periods

is capped at 15). Given that d_t is the first parameter and d_p is the second, the initial formula for price change resistance is:

$$PCR(d_t, d_p) = \left(1 - (1 - d_t/15)^4\right)^{d_p * 40 + 1} \quad (1)$$

This formula was hand-crafted in an attempt to approximate the number of market orders being introduced over several hours following a change in price, as demonstrated in Table III. The formula depends on the magnitude of the change in price, and it is based on subject-matter expert opinion regarding trading behavior. However, we also observed that there tend to be a greater volume in orders at "well-rounded" prices, such as $110, $120, $130, $140, $150, $200, $250, etc. Thus, we applied to the price resistance formula a multiplier that ranges from 1 to 2. The value is one when the nearest "well-rounded" price differs from the current price by at least 1%, and the value is two when the current price is very close to a well-rounded number. We algorithmically make a number between 1 and 10 humanlike by rounding to hundredths if it is less than four or by rounding to twentieths otherwise. For numbers greater than 10, we first divide by 10 until it is no longer greater than 10 (effectively dividing by *exp(floor(log10(n)))*), then perform the above logic, and finally undo the repeated division.

TABLE III. PRICE CHANGE RESISTANCE FORMULA SAMPLES

	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	15%	20%	25%	30%	Price difference
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
1	0.24	0.14	0.08	0.04	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
2	0.44	0.31	0.22	0.16	0.12	0.08	0.06	0.04	0.03	0.02	0.02	0.00	0.00	0.00	0.00	
3	0.59	0.48	0.39	0.31	0.25	0.21	0.17	0.14	0.11	0.09	0.07	0.03	0.01	0.00	0.00	
4	0.71	0.62	0.54	0.47	0.41	0.36	0.31	0.27	0.24	0.21	0.18	0.09	0.05	0.02	0.01	
5	0.80	0.73	0.67	0.62	0.56	0.52	0.47	0.43	0.40	0.36	0.33	0.21	0.14	0.09	0.06	
6	0.87	0.82	0.78	0.74	0.70	0.66	0.62	0.59	0.56	0.53	0.50	0.38	0.29	0.22	0.16	
7	0.92	0.89	0.86	0.83	0.80	0.78	0.75	0.73	0.70	0.68	0.66	0.55	0.47	0.40	0.33	
8	0.95	0.93	0.92	0.90	0.88	0.86	0.85	0.83	0.82	0.80	0.78	0.71	0.65	0.59	0.53	
9	0.97	0.96	0.95	0.94	0.93	0.93	0.92	0.91	0.90	0.89	0.88	0.83	0.79	0.75	0.71	
10	0.99	0.98	0.98	0.97	0.97	0.96	0.96	0.95	0.95	0.94	0.94	0.92	0.89	0.87	0.85	
11	0.99	0.99	0.99	0.99	0.99	0.98	0.98	0.98	0.98	0.98	0.97	0.97	0.96	0.95	0.94	
12	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.98	
13	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
Time																

In addition to these, we included a well-known market indicator, the relative strength index (RSI) [13], as three separate features using different numbers of periods: RSI (14) considers the previous three-and-a-half hours, RSI (480) considers the last five days, and RSI (1344) considers the last two weeks. RSI uses the average price fluctuation in a sliding window to suggest whether the market is overbought (implying it is unlikely that the price will drop much more) or oversold.

3.3 The Genetic Algorithm for Parameter Selection

Because there are so many features and parameters, and individually turning them on and off is not necessarily indicative of their usefulness, we decided to employ an evolutionary strategy to vary many of the parameters. We ran numerous trials with populations of 30 to 100 chromosomes and 50 to 560 generations, also adjusting the probability of mutation and cross-breeding and some other details along the way, but the final strategy is as follows.

First, generate an initial population of seventy individuals by taking three predefined chromosomes and cross-breeding them with an "anti-default" chromosome (in which booleans are toggled from the defaults and numbers are set to one extreme of the allowed mutation range) until the desired population size of 100 chromosomes is reached. Then, separately for each target cryptocurrency, generate the features, perform machine learning, and evaluate each chromosome. Once the current population is evaluated, select the best one for each cryptocurrency (without selecting duplicate chromosomes), and add one clone and one mutation of each selected chromosome into the new population, which is initially empty. Then sort the remaining old population by the sum of scores across all cryptocurrencies and remove the worst 50% of them. Until the new population reaches the desired size of 100 chromosomes, randomly select and perform one of the following sets of operations and remove the used chromosomes from the list:

1. Cross-breed two chromosomes to produce four offspring (60% chance)

2. Mutate one chromosome to produce one offspring and mutate it again to produce a second (20% chance)

3. Clone one chromosome to produce one offspring and mutate it to produce a second (20% chance)

This differs from the standard genetic algorithm in that a randomly-chosen action may have more than one genetic operator. After eighty generations, and only the feature scales are mutated, while other possibilities remain unchanged. In either case, the next step is to remove the last-added one or two as needed in order to keep the population size constant. The process repeats until the desired number of generations (one-hundred) have been evaluated. See Appendix B for pseudocode of this algorithm.

As shown in Table IV, the chromosome is comprised of nine booleans and effectively twenty-six categorical settings (three non-numeric, three integers with limited range, and twenty floating points with limited range and steps). Excluding feature scales, there are 248,371,200 unique chromosomes; however, cluster count only affects the k-means learning method.

It is worth noting that using the profit from a simulation could lead to worse generalization, as this is treating the test data range as training for the genetic algorithm. However, we believe the effect should be negligible because the inner machine learning algorithm is not trained on the test data range. Also, other options for scoring showed little promise for translating into usefulness in trading.

We centered the range for the price rise and drop thresholds around the results of a higher-precision brute force search for the thresholds that yield the greatest profit in the case of perfect 15-minute look-ahead prediction, which identified the optimal thresholds to be +0.3% and -0.45%, given a transaction fee of 0.3%.

3.4 Dynamicaly Selecting Machine Learning Algorithm

Depending on the chromosome, one of three machine learning algorithms is executed on the training set. The result is then used to predict whether each time period in the test set is a large price drop, large price rise, or neither, and those

predictions are passed off to the scoring mechanism as shown in Figure 1. This process is performed separately for each of the target digital currencies, and some specifics depend on the selected machine learning algorithm.

The first machine learning algorithm is a support vector machine (SVM) using stochastic gradient descent [22]. Actually, one SVM is trained to differentiate whether the price increases by at least the price rise threshold specified in the chromosome or not, and a second SVM learns to differentiate whether the price will drop more than the price drop threshold or not. If both SVMs predict a significant change in the price, the conflict is resolved by changing the final prediction to "no change."

TABLE IV. CHROMOSOM DESCRIPTION

Gene Description	Possible Values
Learning method	SVM, k-means, linear regression
Correlation method	generic, daily, weekly
Trading strategy	"All-in, all-out" or "Half-in, half-out"
Include the last X periods' price changes	0, 1, 2, 3, 4
Include the first X cross-correlation results	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Include RSI (14)	false, true
Include RSI (480)	false, true
Include RSI (1344)	false, true
Include weekday price change probabilities	false, true
Include time-of-day price change probabilities	false, true
Include previous period trading volume	false, true
Include 24-hour price change	false, true
Include pay day heuristic	false, true
Include price resistance heuristic	false, true
Cluster count (for k-means learning method)	2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Price rise threshold	1.0015 to 1.0045, step 0.0005
Price drop threshold	0.994 to 0.997, step 0.0005
Feature scales (x18)	0.1 to 1.0, step 0.1

The second machine learning algorithm is k-means clustering [23]. After the clustering completes, we assign each cluster a category (rise, drop, or neither) according to the majority of the data points in that cluster. For example, if a cluster contains the data for 60 intervals that involved price drops and 20 that did not, any data point that fits into that cluster will be predicted as a price drop.

The third machine learning algorithm is linear regression using the ordinary least squares method [24]. After performing linear regression, we convert the results into categories (rise, drop, or neither) by applying the chromosome's selected thresholds.

3.5 Scoring

The score is simply the percent profit, measured in cryptocurrency units gained through trading after the first purchase, based on a simulation executed on the test set. By design, a score of 0 is equivalent to a trader making a purchase at the first predicted price rise and then making no more trades; we felt this was the most sensible baseline, as it represents a lucky individual using the simplest trading strategy (commonly described online as "hold on for dear life"). The fee is picked depending on the cryptocurrency--0.25% for Bitcoin and 0.3% for the others. Price slack is not considered in the simulation, making it less accurate for larger capital investments.

The following steps describe the trading strategies we implemented, but with an exception for the final time period:

1. If a price rise and a price drop are both predicted for the same interval, do nothing.

2. If a price drop is predicted, sell all held cryptocurrency in a taker order.

3. If a price rise is predicted, use all held fiat currency to purchase cryptocurrency in a taker order.

In the final interval, if no cryptocurrency is held, perform step 3 for the sake of scoring; this reduces the score by a factor of 0.003 (0.0025 for Bitcoin). At this stage, the software also calculates what the profit would be if one followed the same trading strategy with perfect prediction accuracy, as the maximum possible profit depends on the price rise and drop thresholds.

We refer to the exact trading strategy above as "all-in." We developed a second trading strategy, "half-in," which only differs in that only half of the available fiat currency is spent when no cryptocurrency is held and only half of the available cryptocurrency is sold when no fiat currency is held. Note that when both fiat and digital currencies are held, the half-in strategy behaves the same as all-in.

3.6 Objective Analysis

Objective analysis could be a completely different research on its own, so many of the details are mere expert opinion, approximation, and guesswork, but we thought it could be very helpful to include anyway. The objective analysis engine is comprised of language-specific logic and a few word-to-value maps with different purposes (the details are presented in an upcoming journal article). Both the English and Japanese analyzers have a word-to-polarity map (much like sentiment analysis, albeit with different words) and a word-to-market-share map (a list of primarily country names), as well as methods of determining negation and news recency from the phrasing, but the exact implementation differs. Regardless of the language, the market share map's values are calculated for each day based on trading volume between cryptocurrency and the primary fiat currency associated with that word. Should no

nation be mentioned in a tweet, the market share factor defaults to 0.5, as both Japan and America have had approximately equal trading volume historically. Similarly, words and grammatical constructs can hint at whether the tweet is referring to a past, present, planned, or hypothetical event, but if no such words or constructs are found, the recency weight defaults to 0.4 (for English) or 0.3 (for Japanese).

A quick glance at side-by-side news volume and price charts suggests that news volume correlates with price increases, so we chose to set the starting total tweet polarity to a very small positive value (0.0005). Any time a word is found in the word polarity map with a negative that appears to apply to it, its polarity is subtracted from the total tweet polarity; if no negative word is near the word, the polarity is added to the total instead. For both languages, encountering a question mark results in the polarity being reduced to 5% of its previous value, based on the likelihood that a question is actually news.

The market share is adjusted by a function which is intended to give more impact to nations that are less active in the market because (we believe) a significant portion of traders are likely to respond to any news rather than only responding to news that applies directly to their own nation. This Adjusted Market Share Factor, or AMSF, is illustrated in equation 2, where *s* represents the market share.

$$\mathrm{AMSF(s)} = (-\log_{10}(\mathrm{s} + 0.00000001) + 1) * \mathrm{s} \quad (2)$$

Each tweet's total impact is calculated as the product of total tweet polarity (*i*), recency weight (*r*), and adjusted market share factor (*AMSF(s)*) as shown in equation 3.

$$\mathrm{i} = \mathrm{p} * \mathrm{r} * \mathrm{AMSF(s)} \quad (3)$$

4 Results

For the final results, we ran six trials, with several using different date ranges. Note that tweets were only available for November 8 through March 28. Every trial involved executing 100 generations of 70 chromosomes with mutations in the last 20 generations restricted to only adjusting weights (cloning and crossover were still possible, and probabilities did not change). In each trial, a winning chromosome was identified and recorded for every digital currency. The first trial was given six winners of flawed past trials as inputs for initialization, while subsequent trials included all prior winners (Table V).

Each trial resulted in at least one chromosome that was profitable for multiple digital currencies and at least one chromosome that obtained over 80% profit in a simulation on the test set; the one trial performed with an 8-week test set achieved over 500% profit. We were reluctant to include prediction accuracy numbers in the results for a few reasons. First, always predicting that the price will not change significantly results in greater than 50% accuracy, and this baseline grows with the price rise and drop thresholds. Second, predicting a large gain when there is a large drop or predicting a large drop when there is a large gain is usually much worse than predicting a gain or drop when the price barely fluctuates, but (third) the impact of failing to predict a large gain, which could be as little as 0.3% or even more than 1%, depends on the situation. Finally, because the rise and drop thresholds are part of the chromosome, no raw accuracy measure we considered was necessarily representative of usefulness for trading.

TABLE V. TRIALS

Training start	Training end / test start	Test end	Training Intervals	Test Intervals	Chromosome ID range
05/15/2017 00:30	03/02/2018 11:45	03/30/2018 11:30	27,980	2,688	7026-7030
05/15/2017 00:30	01/24/2018 00:15	03/21/2018 00:00	24,382	5,376	14103-14107
05/15/2017 00:30	02/21/2018 00:15	03/21/2018 00:00	27,070	2,688	21188-21192
05/15/2017 00:30	02/09/2018 00:15	03/09/2018 00:00	25,918	2,688	28241-28245
05/15/2017 00:30	02/09/2018 00:15	03/09/2018 00:00	25,918	2,688	35320-35324
05/15/2017 00:30	02/09/2018 00:15	03/09/2018 00:00	25,918	2,688	42389-42393

We considered mean square error, but there is no reason to consider a gain significantly larger than the threshold to contribute to an error measure. Thus, we developed a modified version of mean squared error in which the error is fixed at 0 for a prediction that is categorically correct, but if the category is incorrect, the error is calculated as the difference between the category threshold and the actual change in price. This modified formula is represented by equation 4, where *C* is the sequence of predicted classifications, t_+ is the price rise threshold for that chromosome, t_- is its price drop threshold, *p* is the actual percent change in market price for that period, and t_c is the price rise or drop threshold selected according to the prediction for that period.

$$\mathrm{PMSE(C)} = \frac{1}{|C| \times (t_+ - t_-)^2} \sum_{c \in C} \begin{cases} 0 & \text{if categorically correct} \\ (\mathrm{p} - \mathrm{t_c})^2 & \text{if categorically incorrect} \end{cases} \quad (4)$$

The divisor was intended to counter the flaw that smaller thresholds result in smaller average error for the same predictions, even though the smaller thresholds are less useful for trading (especially due to fees and price slack).

Based on the range of profits among the winning chromosomes, it appears that the results may generalize. We can consider how the winning chromosomes differ between trials as a hint as to how well the results will generalize and for how long success may be expected without executing the genetic algorithm again. Table VI shows, in the first row, the most common value for each parameter of the chromosomes; in other rows, cells are left empty for easier comparison if they would have the same value as the first row.

We can see that most parameters (such as the machine learning algorithm, the trading strategy, and the exclusion of the historical average rise/drop for that time of day) are rather consistent among the winning chromosomes, while a few (such as price rise threshold and which versions of RSI to include) are points of contention. Features that appear in fewer winning chromosomes are less likely to be helpful when generalizing.

Excluding the trial with the 8-week test set, the winning chromosomes for Bitcoin range from 78% to 92% profit. Similarly, Ethereum's results range from 28% to 105%, and Litecoin's from 82% to 122%. With perfect prediction, these profit percentages would range in the thousands. However, one should realize that large profit percentages such as these are

unsustainable because the user's effect on the market becomes greater as (s)he trades in increasing quantities. To combat this increasing effect, each purchase should be made with the same amount of fiat currency, once that amount becomes large enough to cause significant price slack. This strategy virtually holds constant the price slack relative to the price change resistance heuristic. Because historical order book data is unavailable, we were unable to account for price slack in simulations, but we believe based on subject-matter expert opinion that trades worth as much as $50,000 do not tend to cause price slack even for the digital currency with the lowest total market capitalization in this study, Litecoin.

TABLE VI. CHROMOSOMES COMPARED TO MOST COMMON VALUES

Chromosome	Machine Learning Algorithm	Trading Strategy	Price Rise Threshold	Price Drop Threshold	Cross-Correlation Method	Tweet Correlations	Previous Candles	RSI	Day Avg	Time Avg	Volume	Pay Day	24h Change	Resistance
(Base)	SVM	All-In	0.003	-0.0025	Weekly	1	1	480	Y	N	Y	Y	Y	Y
7026							3				N			
7028						8	3							
7030			0.0035			9		14, 1344						
14103							3	14			N	N		
14105	Regression		0.004	-0.004				None				N		
14107	Regression		0.004	-0.004				14				N		N
21188			0.004	-0.0035			3	1344			N			
21190		Half-In			Generic	5	3				N			
21192						5	3	1344			N			
28241			0.004	-0.0035				14, 1344	N					
28243			0.004		Daily	9						N	N	
28245					Daily				N					
35320			0.0035	-0.0035				14, 1344						
35322	KMeans (11)		0.004		Generic	9						N	N	
35324					Daily				N					
42389			0.0035	-0.0035				14, 1344						
42391			0.004		Daily	9						N	N	
42393					Daily				N					

5 Conclusion and Future Work

This study developed and evaluated numerous inputs with different machine learning algorithms in order to predict price fluctuations in digital currencies for the sake of real-time trading. It showed that prediction is possible to a large enough extent to yield decent profits. It also showed that the inclusion of heuristics and tweet analysis was helpful in making accurate predictions.

Many other possible improvements remain to be evaluated, both in design and in implementation. Using the candles' high and low prices instead of opening or closing prices may allow the use of maker orders, which could greatly improve profitability by avoiding fees, while simultaneously helping to stabilize the market. More tailored and more accurate sentiment analysis tools might lead to better prediction. Cross-correlation could also be performed on larger offset ranges or time intervals, and if possible, US-based trading data should be isolated so that Daylight Savings time changes would not affect the predictions. The SVM training could be performed differently (perhaps by using a multi-class SVM or the kernel trick); the k-means clustering could be performed repeatedly for the same chromosome to reduce the likelihood of failure due to poor initial groups; and other machine learning methods could be attempted. Even the target classifications could be changed from price rise and price drop to "good time to buy" and "good time to sell" with some effort, which reduces the importance of designing and programming trading strategies manually.

6 References

[1] Coinbase Pro API. https://docs.pro.coinbase.com/

[2] VaderSharp. https://github.com/codingupastorm/vadersharp.

[3] Fallahi, Faraz. (2017). Machine Learning on Big Data for Stock Market Prediction. http://opensiuc.lib.siu.edu/theses/2178/.

[4] Balaji, S. N., Paul, P. V, & Saravanan, R. (2017). Survey on sentiment analysis based stock prediction using big data analytics. In 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), pp. 1–5.

[5] Desokey, E. N., Badr, A., & Hegazy, A. F. (2017). Enhancing stock prediction clustering using K-means with genetic algorithm. In 2017 13th International Computer Engineering Conference (ICENCO), pp. 256–261.

[6] CryptoCompare API. https://www.cryptocompare.com/api/.

[7] Accord.Net. http://accord-framework.net/.

[8] Kravets, Alexander. (2018, January 18). Institutional Investors Will Bet Big on Cryptocurrencies in 2018.

[9] Young, Joseph. (2017, August 11). Institutional Investors Can No Longer Ignore Bitcoin: Goldman Sachs.

[10] Y., Adam. (2017, October 27). How will crypto markets change with the involvement of institutional investors?

[11] Filter realtime Tweets. https://developer.twitter.com/en/docs/tweets/filter-realtime/overview.

[12] VADER. https://github.com/cjhutto/vaderSentiment.

[13] StockCharts. Relative Strength Index. http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:relative_strength_index_rsi

[14] Kamble, R. A. (2017). Short and long term stock trend prediction using decision tree. In 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1371–1375.

[15] Mao, Y., Zhang, Z., & Fan, D. (2016). Hybrid Feature Selection Based on Improved Genetic Algorithm for Stock Prediction. In 2016 6th International Conference on Digital Home (ICDH), pp. 215–220.

[16] Luo, S. S., Weng, Y., Wang, W. W., & Hong, W. X. (2017). L1-regularized logistic regression for event-driven stock market prediction. In 2017 12th International Conference on Computer Science and Education (ICCSE), pp. 536–541.

[17] Vo, N. N. Y., & Xu, G. (2017). The volatility of Bitcoin returns and its correlation to financial markets. In 2017 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC), pp. 1–6.

[18] Shehhi, A. A., Oudah, M., & Aung, Z. (2014). Investigating factors behind choosing a cryptocurrency. In 2014 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1443–1447.

[19] Laskowski, M., & Kim, H. M. (2016). Rapid Prototyping of a Text Mining Application for Cryptocurrency Market Intelligence. In 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), pp. 448–453.

[20] Phillips, R. C., & Gorse, D. (2017). Predicting cryptocurrency price bubbles using social media data and epidemic modelling. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–7.

[21] Cryptocurrency Market Capitalizations. https://coinmarketcap.com/currencies/bitcoin/#markets.

[22] Manning, C. D. (2008). Retrieved from https://nlp.stanford.edu/IR-book/.

[23] Trevino, A. (2016). Introduction to K-means Clustering.

[24] Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1310–1315.