International Journal of Computational Intelligence and Applications
Vol. 19, No. 4 (2020) 2050025 (28 pages)

DOI: 10.1142/S146902682050025X

Prediction of Highly Volatile Cryptocurrency Prices Using Social Media

Mason McCoy

Department of Computer Science
Southern Illinois University
Carbondale, IL 62901, USA
masonem99@hotmail.com

Shahram Rahimi*

Department of Computer Science and Engineering
Mississippi State University
Starkville, MS 39762, USA
rahimi@cse.msstate.edu

Received 4 January 2020
Revised 18 May 2020
Accepted 27 July 2020
Published 14 October 2020

Trading cryptocurrencies (digital currencies) are currently performed by applying methods similar to what is applied to the stock market or commodities; however, these algorithms are not necessarily well-suited for predicting cryptocurrency prices. Unlike stock exchanges, which shut down for several hours or days at a time, digital currency prediction and trading seem to be of a more consistent and predictable nature. In this work, we benefit from sentiment analysis of tweets using both an existing sentiment analysis package and a manually tailored "objective analysis," to calculate one impact value for each analysis every 15 min. We then select the most appropriate training method by applying evolutionary techniques and discover the best subset of the generated features to include, as well as other parameters. One of the unique contributions of this work is the analysis of both English and Japanese tweets with a tailored "objective analysis" tool. This resulted in implementation of predictors which yielded 28% to 122% profit in a four-week simulation, much more than simply holding a digital currency for the same period of time.

Keywords: Cryptocurrency; price prediction; meta-learning; genetic algorithms; social media.

1. Introduction

With its value already in the hundreds of billions, digital currency, also known as cryptocurrency, has become a major force in the economy.[1] As a new major player in the market with its own characteristics, the trade of digital currency should not be

*Corresponding author.

executed similarly to the stock market, which is a well-aged mechanism. 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.[2]

Digital currencies 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.[3–5] 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 min 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).

We were able to generate several features that may be useful for predicting price fluctuations. One of our unique contributions is analyzing both English and Japanese tweets with a tailored "objective analysis" tool. This is because the United States and Japan are the two top countries in cryptocurrency trade. We do this separately for each target currency and also include any mentioned nations' digital currency trading volume as a factor, as well as words that indicate recent or old news, but these features are absent from ordinary sentiment analysis techniques. Another unique feature of this study is the inclusion of a price resistance heuristic in hopes of avoiding transactions that would only be profitable if the size was small, as performing larger transactions affects the market price. Additionally, we believe that we are the first to apply a genetic algorithm to digital currency price prediction for the sake of choosing the most fit machine learning algorithm, optimizing the learning parameters, and selecting the best features.

The remainder of this paper is organized as follows. Section 2 presents information one must know to have a complete understanding of the problem, experiments, results, and implications, including brief reviews of related works. Section 3 describes in detail what steps were involved in developing our solution, such as collecting the

data, analyzing it, and comparing results. It also discusses some of the thought processes and experiments involved in designing the solution. Section 4 states and analyzes the results we obtained. Finally, Sec. 5 concludes the paper and identifies many possible avenues of improving the algorithm and software.

2. Background

Much work has been done in machine learning, but only a few works have involved applying it to prediction and trade of digital currency. In order to fully understand this work and its impact, one must first know how digital currency trading is performed, and what has been achieved previously in related works.

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.[6] 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),[7] 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).

Unlike maker orders, a taker order will also cause price slack if there are not enough maker orders at the market price to completely fill it. For example, assume there are maker orders to buy 10 Ethereum using a taker order, five of your Ethereum would sell for $860 each, then the next five will sell for only $850. After this, maker orders for five Ethereum remain at $850, and the market price becomes $857.50. You only receive $8524.35 (98.83% of the original value at market price) in this case. Now consider the case when there are purchase orders for 10 Ethereum at $862.50 and sale orders at $862.51, making the market price $862.505. In this example, you also sell 10 Ethereum using a taker order, and all 10 of them sell for $862.50, leaving you with $8599.13, which is 99.70% of the original value at market price.

We have included Fig. 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

Fig. 1. Coinbase Pro maker orders.

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 *et al.*[8] 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 *et al.*[9] used k-means clustering to obtain satisfactory results in predicting stock prices. Kamble[10] utilized several different market indicators and decision trees to predict short-term stock market trends. Mao *et al.*[11] used a genetic algorithm to select the features to feed into a support vector machine (SVM) for stock market prediction. In Ref. 12, 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[13] predicted Bitcoin prices using SVMs and neural networks. Shehhi *et al.*[14] attempted to identify the factors that give digital currencies their value by performing a survey of only 134 individuals. In Ref. 15, 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[2] used GDELT, a database of global news, in an attempt to predict both stock and Bitcoin prices. Finally, Phillips and Gorse[16] 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

This section begins with a brief overview of the methodology, described as several interconnected components, followed by sections explaining each component in detail. 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 sections below. Figure 2 illustrates the data flow among these components and some of their major constituent parts.

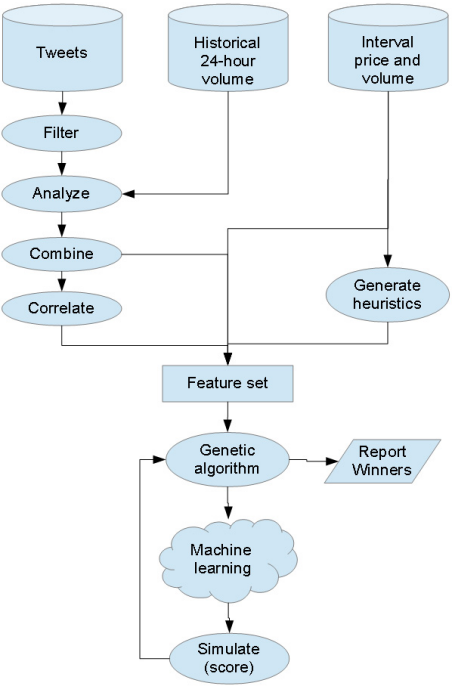

Fig. 2. Data flow diagram.

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[17] for use in the text analysis.

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.[18] 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 1 as inputs for the learning algorithms.

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[19] via k-means clustering, a SVM using stochastic gradient descent, or linear regression using

Table 1. Features for machine learning.

Heuristics	Raw Data
Day-of-week price rise probability	Previous period trading volume
Day-of-week price drop probability	24-h 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)	

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 4 to 12 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 are obtained and saved by three programs to maximize their availability. The first program, a simple Python script developed by one of our peers, receives tweets via Twitter's real-time feed API[20] and saves them in a Mongo database. The second program, GDAXPrices, collects data from the Coinbase Pro API[7] 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 min 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 min 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[17] for use in the objective analyzer.

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 2, while these flagged words are in Table 3. 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

Table 2. Filter flags.

3.2. Flag	3.3. 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

Table 3. Flagged words.

3.4. Word	3.5. Flag	3.6. Word	3.7. Flag
airdrop	AlwaysIgnore	エクスチェンジ	All
エアドップ	AlwaysIgnore	bitcoin	Bitcoin
startup	AlwaysIgnore	ビットコイン	Bitcoin
新興企業	AlwaysIgnore	btc	Bitcoin
ico	AlwaysIgnore	ethereum	Ethereum
altcoin	AlwaysIgnore	エーテル	Ethereum
アルトコイン	AlwaysIgnore	イーサリアム	Ethereum
cryptocurrency	All	eth	Ethereum
暗号侵害	All	litecoin	Litecoin
cryptocurrencies	All	ライトコイン	Litecoin
digital currency	All	ltc	Litecoin
デジタル通貨	All	エクスチェンジ	All
digital currencies	All	bitcoin	Bitcoin
exchange	All		

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[18] for sentiment analysis, and both English and Japanese tweets are given to a separate objective analysis engine for each applicable currency.

NewsChipper executes sentiment analysis using VaderSharp[18] 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."[21] Due to its complexity, objective analysis is separately described later in the paper. For each 15-min 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-min 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-min-period-of-day and 15-min-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 $24 \times 4 = 96$ separate cross-correlation results no matter how many inputs are given, while period-of-week gives $24 \times 4 \times 7 = 672$. 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 0 to 10.

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 (0.01% 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 as follows:

$$\mathrm{PCR}(d_t, d_p) = (1 - (1 - d_t/15)^4)^{d_p \times 40 + 1}. \tag{1}$$

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 4. 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

Table 4. 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																

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.

In addition to these, we included a well-known market indicator, the relative strength index (RSI),[22] as three separate features using different numbers of periods: RSI (14) considers the previous 3½ h, 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.

3.2. *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 –100 chromosomes and 50–560 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 70 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 80 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 (100) have been evaluated. See Appendix A for pseudocode of this algorithm.

As shown in Table 5, the chromosome is comprised of nine Booleans and effectively 26 categorical settings (three non-numeric, three integers with limited range,

Table 5. Chromosome 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-h 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 (×18)	0.1 to 1.0, step 0.1

and 20 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 min look-ahead prediction, which identified the optimal thresholds to be $+0.3\%$ and -0.45%, given a transaction fee of 0.3%.

3.3. *Intelligent selection of the 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 Fig. 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 an SVM using stochastic gradient descent.[23] 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."

The second machine learning algorithm is k-means clustering.[24] 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.[25] After performing linear regression, we convert the results into categories (rise, drop, or neither) by applying the chromosome's selected thresholds.

3.4. *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.

3.5. *Objective analysis*

Objective analysis could be a thesis 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 (see Appendix B). 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 Eq. (2), where s represents the market share.

$$\mathrm{AMSF}(s) = (-\log_{10}(s + 0.00000001) + 1) \times s. \tag{2}$$

Each tweet's total impact is calculated as the product of total tweet polarity, i, recency weight, r, and adjusted market share factor, $\mathrm{AMSF}(s)$, as shown in the following equation:

$$i = p \times r \times \mathrm{AMSF}(s). \tag{3}$$

English. English has spaces and other delimiters that make it easy to split text into individual words, so no particularly interesting logic is required here. The software loops through one character at a time, switching to word identification mode when a delimiter is found after Latin letters. Whether a word's polarity should be negated is determined by the number of words since the last negative word was encountered (if less than five, negate), where a sentence-ending delimiter (a question mark or a period that is not part of an acronym) increases the distance past the threshold. Negative words include not only "not," "cannot," and the suffix "n't," but also some verbs like "reverses" and "backtracks," as well as the preposition "despite". The words "announce," "announces," "new," and "breaking" are treated as recent news (recency weight 1); "plan," "plans," and "intends" are treated as future plans (recency weight 0.8); and "since" and "after" are treated as old news (recency weight 0.2). The chosen polarity words are almost exclusively negative, with the exception of "moon" (as in "Litecoin is going to the moon!"). These words pertain to incidents that seem to cause price drops (bans, hacks, heists, crackdowns, suspending or halting trading, and regulations).

Because we were only able to identify a few words that might correlate with price spikes, we sought insight into what words might have impact by executing a modified version of Nececrus.[5] This software counts words separately for three contexts — within 15 min before a large price increase, within 15 min before a large price decrease, and the rest of the time — that we chose with the thought that market players and individuals posting tweets are probably active at the same time. We converted the word counts into conditional relative probabilities (see Appendix C), but because we were unable to explain the appearance of many of the words with the highest apparent usefulness, we did not utilize this information in the text analyzer. Instead, all polarity scores and recency weights were assigned according to

subject–matter expert opinion, which is based on news stories that coincided with large changes in the cryptocurrency market.

Japanese. Approximately 45% of Bitcoin trading is done between Bitcoin and United States Dollars (USD), while another 45% is done between Bitcoin and Japanese Yen (JPY), according to data from CryptoCompare.[3] Because of this, we wanted to include some analysis of Japanese tweets. The Japanese language differs vastly from English not only in grammar, but also in content. For example, it has several two-letter words that mean "sudden price increase". The word "hacking" is borrowed from English, but almost exclusively appears in its gerund form. There are no spaces to separate words for easy processing, and it becomes impossible to identify the word in some cases without complete knowledge of the surrounding words (such as the word インド, which refers to India, while インドア means "indoor").

However, negatives are attached to the word to which they primarily apply (with either a prefix or a suffix), so it is easy to discover (with greater accuracy than English) when a word with polarity should have the opposite effect. Conjugations are also done almost exclusively by changing the last one letter and/or adding additional letters to the word, whereas English has many exceptions and words that change spelling entirely when conjugated.

One point of interest in the Japanese component is a technical implementation detail. Because the software can have no knowledge of how long a word is until identifying it, we developed a modified binary search that, after it finds one entry that matches the first character of a word, reverts to a linear search leftward, and then rightward to check all the surrounding entries that begin with the same character.

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 6).

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 eight-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

Table 6. 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	2688	7026–7030
05/15/2017 00:30	01/24/2018 00:15	03/21/2018 00:00	24,382	5376	14,103–14,107
05/15/2017 00:30	02/21/2018 00:15	03/21/2018 00:00	27,070	2688	21,188–21,192
05/15/2017 00:30	02/09/2018 00:15	03/09/2018 00:00	25,918	2688	28,241–28,245
05/15/2017 00:30	02/09/2018 00:15	03/09/2018 00:00	25,918	2688	35,320–35,324
05/15/2017 00:30	02/09/2018 00:15	03/09/2018 00:00	25,918	2688	42,389–42,393

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 was considered because such measures would not necessarily indicate the usefulness for trading.

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 in Eq. (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.

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

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). The simulation profit and PMSE values for the winning chromosomes are listed in Table 7.

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 8 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. For readability, feature scales for each chromosome are listed separately in Table 9.

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

Table 7. Chromosome profits and PMSE values.

Chromosome ID	Currency	Profit (%)	Training PMSE	Test PMSE
7026	BTC	92.08	0.745	0.567
7026	ETH	71.60	1.288	0.593
7026	LTC	35.52	1.494	0.802
7028	BTC	17.22	0.639	0.526
7028	ETH	105.28	1.303	0.593
7028	LTC	5.67	1.577	0.812
7030	ETH	19.93	1.026	0.464
7030	LTC	92.78	1.134	0.641
14103	BTC	144.28	0.671	0.882
14103	ETH	203.84	1.711	1.305
14105	BTC	32.94	0.000	0.383
14105	ETH	530.36	0.000	0.285
14105	LTC	220.97	0.000	0.463
14107	BTC	24.34	0.000	0.385
14107	ETH	413.66	0.000	0.288
14107	LTC	271.35	0.000	0.459
21188	BTC	68.47	0.368	0.288
21188	ETH	23.76	0.766	0.329
21190	ETH	70.32	1.777	0.767
21192	ETH	1.82	1.276	0.676
21192	LTC	82.49	1.354	0.878
28241	BTC	77.56	0.365	0.215
28243	BTC	14.18	0.300	0.200
28243	ETH	33.88	0.871	0.217
28243	LTC	18.43	0.175	0.164
28245	LTC	110.97	1.409	0.925
35320	BTC	78.38	0.439	0.260
35322	ETH	28.67	0.000	0.216
35324	LTC	110.97	1.409	0.925
42389	BTC	78.38	0.439	0.260
42391	BTC	14.18	0.300	0.200
42391	ETH	33.88	0.871	0.217
42391	LTC	18.43	0.175	0.164
42393	BTC	14.18	0.801	0.234
42393	LTC	122.58	1.407	0.887

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 eight 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

Table 8. 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	24 h 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					

Table 9. Winning chromosome feature scalers.

Chromosome	Previous Candles (1, 2, 3 ago)	Objective Impact	Sentiment Impact	RSI 14	RSI 480	RSI 1344	Avg. Day Rise	Avg. Day Drop	Previous Volume	Pay Day	24 h Change	Drop Resist	Rise Resist
7026	0.3, 1, 0.1	0.1	0.1	—	0.4	—	0.1	0.8	—	0.7	1	0.5	0
7028	0.1, 1, 0.9	0.4	0.1	—	0.1	—	0.9	1	1	1	0.5	0.1	0
7030	0.1	1	0.1	0.1	—	0.1	0.1	0.1	1	0.6	1	0.9	0.1
14103	0.1, 0.3, 0.1	0.1	0.1	0.1	—	—	0.1	0.1	—	—	1	1	1
14105	0.4	1	0.1	—	—	—	0.6	0.1	0.1	—	1	1	1
14107	0.4	1	0.1	0.6	—	—	0.1	0.1	1	—	1	—	—
21188	0.3, 1, 0.4	0.1	0.1	—	—	0.9	0.6	0.8	—	1	0.1	0.5	0
21190	0.3, 1, 0.8	0.1	0.6	—	0.4	—	0.1	0.8	—	0.7	1	0.8	0.7
21192	0.6, 0.4, 1	1	0.1	—	—	0.4	0.1	0.5	—	0.7	1	0.5	0.3
28241	0.1	0.4	0.3	0.1	—	0.1	—	—	0.5	0.8	1	0.6	0.2
28243	0.1	1	0	—	0.1	—	0.7	0.1	0.1	—	—	0.8	0.5
28245	0.1	1	0	—	0.1	—	—	—	0.9	0.1	0.8	0.7	0.5
35320	0.1	0.4	0.3	0.1	—	0.1	0.5	0.8	1	0.6	0.2	0.1	1
35322	0.1	1	0	—	0.1	—	0.7	0.1	0.1	—	—	0.8	0.8
35324	0.1	1	0	—	0.1	—	—	—	0.9	0.1	0.8	0.7	0.5
42389	0.1	0.4	0.3	0.1	—	0.1	0.5	0.8	1	0.6	0.2	0.1	1
42391	0.1	1	0	—	0.1	—	0.7	0.1	0.1	—	—	0.8	0.5
42393	0.1	1	0	—	0.1	—	—	—	0.9	0.9	0.8	0.7	0.5

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.

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.

The speed of the software leaves plenty of space for real-time execution once a chromosome is selected. In fact, approximately 750 individuals can be trained and evaluated in the interval between real-time predictions (based on the slowest machine learning algorithm we employed, which was also the most successful), so it is also possible to use the genetic algorithm to keep the chromosomes up-to-date as market trends change over time.

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.

Our objective analysis tool, and therefore the prediction accuracy, could likely be improved by repeating a few steps:

(1) Adjust the word impact map based on the conditional probability obtained from Nececrus.
(2) Execute the objective analysis on a larger set of tweets.

(3) Run cross-correlation to determine the time offset when tweets have the most impact on the price.
(4) Adjust Nececrus to count words using the time offset of strongest correlation.

The prediction target could be changed to a greater interval or to a later interval, which may allow making better decisions than only looking at the next 15 min. There are also different trading strategies that might work better, such as trading with more when the prediction is more confident or using maker orders. We did not simulate trading between digital currencies, but that may also yield superior performance.

The training strategy could also be modified. While we chose to train the learning algorithms to classify price changes that meet the gain or loss criterion (which may be adjusted by the genetic algorithm), training the algorithm directly to make optimal trades may produce better results. Trading may also be performed differently depending on the ability to predict larger periods of time or farther into the future.

Historical order book data cannot be directly obtained, but tracking it in real-time would allow us to replace the manually-developed price change resistance heuristics with one obtained via machine learning. Similarly, including recent volumes of units bought and sold separately instead of a combined total volume seems likely to yield better prediction accuracy. If the tweet collection software is disconnected from Twitter for any reason, the missed tweets cannot feasibly be obtained. By running the software in multiple locations, it would be possible to greatly reduce the chances of missing tweets.

Appendix A. Genetic Algorithm Pseudocode

```
PastWinners = a list of three best chromosomes from debugging trials
Default = a default chromosome
AntiDefault = the opposite of the default chromosome
For each trial
    MutationMode = MutateAll
    Population = empty list
    For i = 1 to 70 - PastWinners.Count
        NewChromosome = crossover of Default and AntiDefault
        Add a mutation of NewChromosome to Population
    End For
    Add a clone of each chromosome from PastWinners to Population

    For g = 1 to 100
        If g >= 80, then
            Set MutationMode to MutateWeightsOnly
        End If
        For each chromosome, i, in Population
            Train i
            Score i
        End For

        GenerationWinners = empty list
        For each digital currency, c
            Winner = chromosome in population with highest score for currency c
            If GenerationWinners does not contain Winner, then
                Add Winner to GenerationWinners
            End If
        End For

        For each chromosome, i
            Set i's sort key to the sum of its profits for each currency
        End For
        Sort Population by sort key, descending
        Delete latter half of Population

        NewPopulation = empty list
        For each chromosome, w, in GenerationWinners
            Add a clone of w to NewPopulation
            Add a mutation of w to NewPopulation
        End For
        While NewPopulation has fewer than 70 entries
            Action = generate random number from 0 to 4 inclusive
            If Action is 0, 1, or 2 and Population has at least 2 members, then
                a = randomly select and remove one chromosome from Population
                b = randomly select and remove one chromosome from Population
                Add four crossovers of a and b to NewPopulation
            ElseIf Action is 3, then
                a = randomly select and remove one chromosome from Population
                a = mutation of a
                Add a to NewPopulation
                Add mutation of a to NewPopulation
            Else
                a = randomly select and remove one chromosome from Population
                Add clone of a to NewPopulation
                Add mutation of a to NewPopulation
            End If
        End While
        If NewPopulation has > 70 entries, then
            Remove last-added entry from NewPopulation
        End If
        Population = NewPopulation
    End For
    Add GenerationWinners to PastWinners
End For
```

Appendix B. Objective Analysis Word Maps and Lists

Table B.1. English word-impact map.

Word	Impact (Additive)
ban	−1.0
banned	−1.0
bans	−1.0
banning	−1.0
crackdown	−0.7
crack	−0.7
cracks	−0.7
cracking	−0.7
suspend	−0.7
suspended	−0.7
suspends	−0.7
suspending	−0.7
halt	−0.7
halts	−0.7
halted	−0.7
hacked	−0.5
heist	−0.5
hacks	−0.5
hack	−0.5
regulations	−0.8
moon	+0.3

Table B.2. English nation-currency map.

Word	Currency	Word	• Currency	• Word	• Currency
Japan	JPY	Brazil	BRL	Israeli	NIL
Japanese	JPY	Brazilian	BRL	Africa	NIL
Tokyo	JPY	Canada	CAD	African	NIL
Korea	KRW	Canadian	CAD	Ireland	NIL
Korean	KRW	Mexico	NIL	Irish	NIL
Vietnam	VND	Mexican	NIL	Denmark	NIL
Vietnamese	VND	Indonesia	NIL	Danish	NIL
Poland	PLN	Indonesian	NIL	Philippines	NIL
Polish	PLN	Turkey	NIL	Filipino	NIL
Australia	AUD	Turkish	NIL	Pinoy	NIL
Australian	AUD	Netherlands	NIL	Malaysia	NIL
China	CNY	Dutch	NIL	Malaysian	NIL
Chinese	CNY	Switzerland	NIL	Colombia	NIL
US	USD	Swiss	NIL	Colombian	NIL
U.S.	USD	Saudi	NIL	Singapore	NIL
USA	USD	Argentina	NIL	Singaporean	NIL
U.S.A.	USD	Argentine	NIL	Pakistan	NIL
States	USD	Argentinian	NIL	Pakistani	NIL
America	USD	Taiwan	NIL	Chile	NIL
American	USD	Taiwanese	NIL	Chilean	NIL

Table B.2. (*Continued*)

Word	Currency	Word	• Currency	• Word	• Currency
Britian	GBP	Sweden	NIL	Finland	NIL
British	GBP	Swedish	NIL	Finnish	NIL
UK	GBP	Belgium	NIL	Bangladesh	NIL
U.K.	GBP	Belgian	NIL	Bangladeshi	NIL
Kingdom	GBP	Thailand	NIL	Venezuela	NIL
England	GBP	Thai	NIL	Venezuelan	NIL
European	EUR	Iran	NIL	Portugal	NIL
Germany	EUR	Iranian	NIL	Portuguese	NIL
German	EUR	Austria	NIL	Peru	NIL
France	EUR	Austrian	NIL	Peruvian	NIL
French	EUR	Egypt	NIL	Czech	NIL
Italy	EUR	Egyptian	NIL	Romania	NIL
Italian	EUR	Nigeria	NIL	Romanian	NIL
Spain	EUR	Nigerian	NIL	Greece	NIL
Spanish	EUR	Norway	NIL	Greek	NIL
Russia	RUB	Norwegian	NIL	Zealand	NIL
Russian	RUB	Emirates	NIL	Iraq	NIL
India	INR	Emirati	NIL	Iraqi	NIL
Indian	INR	Israel	NIL		

Table B.3. English word-recency map.

• Word	• Recency (Multiplier)
announce	1
announces	1
new	1
breaking	1
plans	0.8
plan	0.8
intends	0.8
since	0.2
after	0.2

Table B.4. English negative word and suffix list.

no
not
cannot
reverses
backtracks
despite
spite
n't (suffix)
n't (suffix)

Table B.5. Japanese word-impact map.

• Word	• Impact (Additive)	• Meaning
禁止	−1	ban
禁令	−1	ban
解禁	0.8	lift a ban
弾圧	−0.7	crackdown
ハッキング	−0.5	hacking
規制	−0.8	regulations
急騰	0.3	price jump
高騰	0.3	price jump
急伸	0.3	price jump
跳ね上が	0.3	price jump

Table B.6. Japanese nation-currency map.

• Word	• Currency	• Word	• Currency	• Word	• Currency
ジャパン	JPY	ロシア	RUB	イスラエル	NIL
日本	JPY	魯国	RUB	デンマーク	NIL
コリア	KRW	インド	INR	コロンビア	NIL
朝鮮	KRW	ブラジル	BRL	シンガポール	NIL
韓国	KRW	カナダ	CAD	フィリピン	NIL
越南	VND	メキシコ	NIL	パキスタン	NIL
ベトナム	VND	オランダ	NIL	チリ	NIL
ポーランド	PLN	トルコ	NIL	ベネズエラ	NIL
オーストラリア	AUD	スイス	NIL	アイルランド	NIL
チャイナ	CNY	サウジアラビア	NIL	フィンランド	NIL
中国	CNY	アルゼンチン	NIL	バングラデシュ	NIL
アメリカ	USD	台湾	NIL	ポルトガル	NIL
米国	USD	スウェーデン	NIL	ギリシャ	NIL
イギリス	GBP	ナイジェリア	NIL	ペルー	NIL
英国	GBP	ベルギー	NIL	カタール	NIL
ドイツ	EUR	ノルウェー	NIL	チェコ	NIL
独国	EUR	イラン	NIL	ルーマニア	NIL
フランス	EUR	アラブ	NIL	カザフスタン	NIL
仏国	EUR	エジプト	NIL	アルジェリア	NIL
イタリア	EUR	南アフリカ	NIL	ニュージーランド	NIL
スペイン	EUR	マレーシア	NIL	イラク	NIL

Table B.7. Japanese word-recency map.

• Word	• Recency (Multiplier)	• Meaning
公表	0.9	official announcement
発表	0.9	announcement
予定	0.7	plan
報告	0.8	report
知らせ	0.8	news

Table B.8. Japanese negative word and prefix list.

解く	解か	解け
解除		
無い	ない	
無く	なく	
無し	なし	
無 (prefix)	不 (prefix)	

Table B.9. Backup currency pair-market share map.

Currency Pair	Market Share	Currency Pair	Market Share	Currency Pair	Market Share
BTCJPY	0.4602	ETHUSD	0.2928	LTCUSD	0.3016
BTCUSD	0.4198	ETHGBP	0.1231	LTCJPY	0.1837
BTCEUR	0.0542	ETHRUB	0.1151	LTCGBP	0.1154
BTCKRW	0.0439	ETHBRL	0.1151	LTCRUB	0.1152
BTCVND	0.0084	ETHCNY	0.1150	LTCCNY	0.1150
BTCGBP	0.0060	ETHINR	0.1150	LTCINR	0.1150
BTCCAD	0.0021	ETHJPY	0.0557	LTCEUR	0.0265
BTCPLN	0.0018	ETHEUR	0.0374	LTCKRW	0.0203
BTCAUD	0.0015	ETHKRW	0.0244	LTCVND	0.0034
BTCRUB	0.0011	ETHVND	0.0032	LTCAUD	0.0015
BTCBRL	0.0007	ETHAUD	0.0013	LTCCAD	0.0012
BTCCNY	0.0002	ETHCAD	0.0012	LTCPLN	0.0007
BTCINR	0.0001	ETHPLN	0.0007	LTCBRL	0.0006
BTCETH	0.0000	ETHBTC	0.0000	LTCBTC	0.0000
BTCLTC	0.0000	ETHLTC	0.0000	LTCETH	0.0000
BTCUSDT	0.0000	ETHUSDT	0.0000	LTCUSDT	0.0000
BTCNIL	0.0000	ETHNIL	0.0000	LTCNIL	0.0000

Appendix C. World Related Frequencies from Nececrus

Table C.1. Words with higher conditional probability.

• Before a Price Rise		• Before a Price Drop	
• Word	• Rise/Neutral Frequency	• Word	• Drop/Neutral Frequency
panic	1.612	competition	3.035
protects	1.469	lists	2.965
crash	1.465	lightning	2.882
inside	1.458	february	2.851
levels	1.441	science	2.803
passed	1.408	scams	2.792
drops	1.371	marketplace	2.664
aka	1.365	countries	2.637
kucoin	1.355	freedom	2.573
freedom	1.355	coinmetro	2.517
ban	1.334	faucet	2.481

Table C.1. (*Continued*)

• Before a Price Rise		• Before a Price Drop	
• Word	• Rise/Neutral Frequency	• Word	• Drop/Neutral Frequency
developing	1.331	totally	2.475
passes	1.323	whitelist	2.439
competition	1.318	developing	2.420
random	1.313	management	2.406

Table C.2. Words with lower conditional probability.

• Before a Price Rise		• Before a Price Drop	
• Word	• Rise/Neutral Frequency	• Word	• Drop/Neutral Frequency
ppkc	0.468	investy	0.000
plays	0.532	vida	0.001
numbers	0.592	ibmauto	0.041
betting	0.678	christmas	0.065
guy	0.688	cboe	0.100
campaign	0.707	nov	0.114
tim	0.717	plays	0.165
gave	0.737	ces	0.165
vip	0.764	comext	0.174
became	0.783	pick	0.197
va	0.791	december	0.226
pitch	0.796	cme	0.246
ibmauto	0.842	gave	0.250
else	0.849	reports	0.269
blockchains	0.863	cryptocurrent	0.269

Table C.3. Potential best direction predictors.

• Price Rise Predictors		• Price Drop Predictors	
• Word	• Rise/Drop Frequency	• Word	• Rise/Drop Frequency
Investy	2023.956	science	0.352
vida	1249.334	marketplace	0.418
ibmauto	20.630	lists	0.428
christmas	13.938	coinmetro	0.433
cboe	11.732	competition	0.434
nov	8.234	ecosystem	0.437
pick	6.257	lino	0.437
ces	6.023	february	0.444
comext	5.716	lightning	0.444
december	4.847	scams	0.448
passes	4.828	faucet	0.450
reports	4.380	viberate	0.457
cme	3.971	officially	0.458
bought	3.744	whitelist	0.461
dec	3.726	generation	0.462

References

1. M. McCoy and S. Rahimi, Towards a Twitter-based prediction tool for digital currency, in *Proc. 21st Int. Conf. Artif. Intell. ICAI'19*, Las Vegas, USA, 2019, pp. 305–311.
2. F. Fallahi, Machine learning on big data for stock market prediction, thesis, Southern Illinois University (2017).
3. A. Kravets, *Institutional Investors Will Bet Big on Cryptocurrencies in 2018* (Cointelegraph, 2018).
4. J. Young, *Institutional Investors Can No Longer Ignore Bitcoin: Goldman Sachs* (August 11, 2017). https://cointelegraph.com/news/institutional-investors-can-no-longer-ignore-bitcoin-goldman-sachs.
5. Y. Adam, *How will Crypto Markets Change with the Involvement of Institutional Investors?* (October 27, 2017). https://hackernoon.com/how-will-crypto-markets-change-with-the-involvement-of-institutional-investors-a7808cabed99.
6. J. Tuwiner, How to Buy Bitcoins with Cash or Cash Deposit (July 25, 2019), https://www.buybitcoinworldwide.com/en/buy-bitcoins-with-cash.
7. Coinbase Pro API (2018), https://docs.pro.coinbase.com/.
8. S. N. Balaji, P. V. Paul and R. Saravanan, Survey on sentiment analysis-based stock prediction using big data analytics, in *Proc. Innov. Power Adv. Comput. Technol.-i-PACT*, Vellore, India, 2017, pp. 1–5.
9. E. N. Desokey, A. Badr and A. F. Hegazy, Enhancing stock prediction clustering using K-means with genetic algorithm, in *Proc. 13th Int. Comput. Eng. Conf. ICENCO*, Cairo, 2017 (Cairo University Giza, Egypt, 2017), pp. 256–261.
10. R. A. Kamble, Short and long term stock trend prediction using decision tree, in *Proc. 2017 Int. Conf. Intelligent Computing and Control Systems*, New Jersey, USA, 2017, pp. 1371–1375.
11. Y. Mao, Z. Zhang and D. Fan, Hybrid feature selection based on improved genetic algorithm for stock prediction, in *Proc. 6th Int. Conf. Digital Home-ICDH*, Guangzhou, China, 2016, pp. 215–220.
12. S. S. Luo, Y. Weng, W. W. Wang and W. X. Hong, L1-regularized logistic regression for event-driven stock market prediction, in *Proc. 12th Int. Conf. Computer Science and Education-ICCSE*, Houston, USA, 2017, pp. 536–541.
13. N. N. Y. Vo and G. Xu, The volatility of Bitcoin returns and its correlation to financial markets, in *Proc. Int. Conf. Behavioral, Economic, Socio-cultural Computing-BESC*, Krakow, Poland, 2017, pp. 1–6.
14. A. A. Shehhi, M. Oudah and Z. Aung, Investigating factors behind choosing a cryptocurrency, in *Proc. IEEE Int. Conf. Industrial Engineering and Engineering Management*, Selangor Darul Ehsan, Malaysia, 2014, pp. 1443–1447.
15. M. Laskowski and H. M. Kim, Rapid prototyping of a text mining application for cryptocurrency market intelligence, in *Proc. IEEE 17th Int. Conf. Information Reuse and Integration-IRI*, Pittsburgh, USA, 2016, pp. 448–453.
16. R. C. Phillips and D. Gorse, Predicting cryptocurrency price bubbles using social media data and epidemic modelling, in *Proc. IEEE Symp. Series Computational Intelligence-SSCI*, Hawaii, USA, 2017, pp. 1–7.
17. CryptoCompare API (April 10, 2017), https://www.cryptocompare.com/api/.
18. VaderSharp (2017), https://github.com/codingupastorm/vadersharp.
19. Accord.Net (April 10, 2017), http://accord-framework.net/.
20. Filter Realtime Tweets (May, 2017), https://developer.twitter.com/en/docs/tweets/filter-realtime/overview.

21. VADER (January 2018), https://github.com/cjhutto/vaderSentiment.
22. StockCharts- Relative Strength Index (January, 2018), http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:relative_strength_index_rsi.
23. C. D. Manning (January 2018), Retrieved from https://nlp.stanford.edu/IR-book/.
24. J. Wu, *Advances in K-means Clustering* (Springer, Berlin Heidelberg, 2014).
25. A. Singh, N. Thakur and A. Sharma, A review of supervised machine learning algorithms, in *Proc. 3rd Int. Conf. Computing for Sustainable Global Development- INDIACom*, New Delhi, India, 2016, pp. 1310–1315.