CS230

Cryptocurrency Price Change Prediction Using News and Social Media Sentiment

Connor Lamon
Stanford University
conlamon@stanford.edu

Eric Nielsen
Stanford University
nielsene@stanford.edu

Eric Redondo
Stanford University
eredondo@stanford.edu

Abstract

As the economic and social impacts of cryptocurrencies continue to grow, so does the prevalence of related news articles and social media posts. Similar to traditional financial markets, there appears to be a relationship between media sentiment and the prices of cryptocurrency coins. This project analyzes the ability of news and social media data, combined with the power of Recurrent Neural Networks, to predict Bitcoin price fluctuations over time periods ranging from 1 to 24 hours in the future. Rather than label news and social media text on perceived sentiment, data was labeled on the actual price changes a set amount of time in the future, allowing the model to directly predict price fluctuations. The model consisted of an embedding layer utilizing a custom Word2Vec model, a single bi-directional LSTM layer, and a final linear activation layer to predict price changes. Overall prediction accuracy was highest (54.5%) using a model trained on Reddit posts from relevant crypto communities when predicting price changes 12 hours in the future.

1 Introduction

This project served to answer two questions: First, can sentiment analysis on news headlines, tweets, and Reddit posts produce accurate predictions on the future price changes of Bitcoin 1 hour, 2 hours, 6 hours, 12 hours and 24 hours in the future? Second, if predictions from each data set are possible, which is the better indicator of future Bitcoin prices?

The input to our system is text phrases from news headlines, tweets and Reddit posts, kept in order of occurrence to preserve the time-series nature of the data. A regression task is then performed using a custom Word2Vec embedding layer followed by a Recurrent Neural Network (RNN), which produces numerical predictions of the percent change in future Bitcoin prices based on each piece of text separately. All of the outputs are then aggregated by hour and a final prediction is made using the average prediction value for that given hour.

While the system shows potential to directly predict price changes, it may be more useful in detecting whether or not a given headline or media post contains text that has shown to be relevant to changes in price in the past.

2 Related work

The prospect that online sentiment can be measured and used to predict market price changes has been studied extensively for the traditional stock market. Attempts such as those by Pagolu et. al [1] have utilized traditional supervised machine learning algorithms to demonstrate the correlation between sentiment in Twitter data and the Dow Jones Industrial Average (DJIA). More notably, Bollen et. al [2] were able to achieve 87% accuracy in predicting whether the price of the DJIA would go up or down on a daily time period. This method involved using solely tweets and a clustering neural network known as a self-organizing fuzzy neural network.

There have also been previous attempts to utilize sentiment from tweets to predict fluctuations in the price of Bitcoin. Stenqvist and Lonno [3] utilized a rule based method of determining sentiment in tweets and then aggregated sentiment on a slightly higher frequency time scale of every 30 minutes. They achieved 79% accuracy in predicting bitcoin price fluctuations using 2.27 million tweets. Their method was useful in deciding where to start. The approach of using historical price data of cryptocurrencies to make predictions has also been explored. Jiang and Liang [5] utilized deep reinforcement learning and price data to manage a Bitcoin portfolio that made a 10x gain in portfolio value. Finally, Shah and Zhang [4] utilized Bayesian regression to double their investment over a 60 day period. None of these methods utilized news or social media data to capture trends not apparent in the price history data.

This project is differentiated from the projects outlined above through the different sources of media data, the method of labeling the data based on direct price change, the form of percent change analysis, and the use of advanced deep learning methods.

3 Dataset and Features

Data required for the model comes in two forms - technical market data and online media. First, historical hourly Bitcoin price data was obtained using a script that scraped data from Bitcoincharts [6]. Second, three other online scraping tools were built to acquire ∼4,000 cryptocurrency-related news article headlines from Coindesk [7], ∼100,000 cryptocurrency-related tweets (using keyword searches and targeted Twitter accounts) [9], and ∼100,000 posts from prominent cryptocurrency-related subreddits (e.g., r/bitcoin, r/cryptocurrencies, r/cryptomarkets) [8]. The data acquired extends from January 1, 2017 to March 6, 2018.

The historical Bitcoin prices were used to label each text data point with five values: the actual percent changes in Bitcoin price at 1 hour, 2 hours, 6 hours, 12 hours, and 24 hours after the publication of the text (beginning at the top of the next hour). An example can be seen in Table 1. The labeled data was then segmented into train/dev and test sets, using an 80%/20% split for the smaller headline data set and a 90%/10% split for the larger tweets and Reddit posts data sets.

Table 1: Example Labeled News Headline

Timestamp	Headline
2017-07-28 19:10:13	Bitcoin’s market capitalization relative to the rest of the cryptocurrency market has risen above 50%

+1hr Change	+2hr Change	+6hr Change	+12hr Change	+24hr Change
0.91130	1.22585	-0.76038	-0.20546	-0.79166

For early versions of the model, historical daily price data was used for Bitcoin, Ethereum, and Litecoin, obtained with a script that used the CryptoCompare API [10]. Text data was labeled with binary 1/0 values based on the direction of price change (positive or negative) one day after text publication. As will be discussed later, the strategy changed to focus on hourly price data and non-binary labels after initial experimentation results.

The text data was pre-processed in three steps. First, web links had the ’http’ portion removed. Second, all of the text was made to be lowercase. Finally, all characters that were not alpha-numeric were removed. Numbers were not removed as they were found to be useful and relevant to cryptocurrency sentiment. Features were generated by training a custom Word2Vec model utilizing all of the ∼200,000 text samples. This model was trained using the skip-gram approach and the gensim python library [16]. Initially, the standard Google pre-trained Word2Vec model [15] was used, but it was found that this model had not been trained on many critical cryptocurrency words and phrases, such as ’bitcoin’. The custom model, even with only ∼200,000 text samples, was able to recognize crypto semantics and slang words. For example, the closest (cosine distance) vector to "ico" was found to be "tokensale", and the closest vector to "mining" was "rig"; see Table 2. Additionally, the model was able to complete analogies such as: "bitcoin" is to "eth" as "ethereum" is to "btc". The final, and most humorous, example involved the closest vector to the term "hodl", which was "btfd". The reader is advised to web search for these slang terms if unfamiliar.

Table 2: Query Top Four Similar Words to *'mining'* using Crypto Word2vec model

Similar Word	rig	pool	mine	gpu
Cosine Distance	0.648	0.635	0.592	0.582

4 Methods

All of the models were implemented using Keras, with Tensorflow as the backend. Initially, nine binary classification models were created - one for each of the three data sources - to make binary predictions on the direction of future price changes for three coins: Bitcoin, Ethereum, and Litecoin. However, the final implementation instead involved three regression models - one for each data source to make predictions on the actual future price changes for Bitcoin.

4.1 Initial Models

Each of the nine baseline models was constructed using a similar design which involved passing input data through three layers: an embedding layer, an RNN layer, and a sigmoid activation layer. Words in each input text phrase were first tokenized using a vocabulary limited to 20,000 unique words, and then were passed through an embedding layer to encode each word using a 300-dimensional vector representation. The embedding layers, for these models, were created using Google's pre-trained Word2vec model previously discussed. The encoded words were then passed through an RNN with a single layer comprised of 128 bi-directional long short-term memory (LSTM) blocks. By using a bi-directional scheme, each block is able to reason about the current word using information learned about words coming both before and after it in the phrase. The extent to which preceding and trailing words influence the current word's output is controlled by equations 1, 2, and 3. The output of the sigmoid layer was interpreted as a 1 or a 0, indicating a price increase or decrease, respectively.

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u) \tag{1}$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f) \tag{2}$$

$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o) \tag{3}$$

4.2 Final Models

Each of the three final models was constructed using a pipeline with the same three types of layers as the baseline, but the designs of the embedding layer and activation layer were updated.

The embedding layer was created using the custom, crypto-specific Word2Vec model described previously. Tokenized words from each input text phrase were passed through this layer and transformed into 200-dimensional vectors. Each encoded word was then passed through the same 128 block bi-directional LSTM layer.

Since the prediction goal was updated to be the actual price change percentages rather than the binary price change directions, the dense output activation layer was modified to use a linear activation function. Also, since predictions were desired for multiple time periods in the future, the activation layer was changed to output a 5-dimensional vector with values corresponding to each of the 5 prediction times (1 hour, 2 hours, 6 hours, 12 hours, and 24 hours in the future).

Final models were also trained using mini-batches and the Adam optimization algorithm, but minimized a different loss function: mean-squared error. This loss function is standard for regression tasks and is defined as follows:

$$Loss = \frac{1}{m}\sum_{1}^{m}(y_m - \hat{y_m})^2 \tag{4}$$

5 Results and Discussion

5.1 Initial Models and Design Updates

For the initial models, overall prediction accuracy was the evaluation metric, since it mattered equally to correctly predict a 0 (price decrease) or 1 (price increase). Using this metric, as shown in Table 3 below, it was clear that the models were overfitting the training set, and as a result, were performing poorly on the test set. In order to better evaluate each model, the magnitude of price changes being predicted correctly and incorrectly was examined. There appeared to be little correlation between the magnitude of price changes and the accuracy of the predictions.

Table 3: BTC +1 day Prediction Accuracy Using a Single Layer Bi-Directional LSTM

	News Headlines	Tweets	Reddit Posts
Train Accuracy	99.2%	88.52%	87.34%
Test Accuracy	52.5%	49.70%	50.55%

Given the poor results of the initial models, it was decided to make several substantial changes to the design. First, the choice was made to increase the granularity of prediction times in an attempt to better capture the correlation between text publication and price changes. This involved obtaining a new set of price data with values for every hour rather than every day. Unfortunately, only historical hourly data for Bitcoin could be obtained, so predictions on Ethereum and Litecoin price were dropped from the scope of the model. Second, it was decided to increase the expressiveness of the predictions by labeling data points with actual percent changes in coin price rather than a binary indicator of price change direction. The intention was to give the model more flexibility in differentiating between different pieces of text. Third, the embedding layer was changed to use a custom word2vec model, as previusly discussed. Finally, input data used for the Reddit and Twitter models was filtered to contain fewer irrelevant/spam posts. For the Reddit model, only posts with overall upvote scores of at least 10 were considered. For the Twitter model, only tweets produced by one of ∼20 core Bitcoin developers were considered.

5.2 Final Models

During the training of the final models, there were multiple hyperparameter choices that had to be made. For training the word2vec model, the hyperparameters involved the minimum count a word had to appear in the entire text corpus to be considered in the model, and the vector dimension for representing each word. These were chosen to be 5 and 200 respectively, through experimentation. Both were optimized to produce the best results while minimizing memory consumption. Next, the LSTM model hyperparameters included choosing the mini-batch size and the optimization algorithm parameters. The mini-batch size was chosen to be 32 as it was the largest mini-batch size that would still run in a reasonable time. The Adam optimization algorithm was chosen due to its learning speed and it was used with a standard learning rate of 1e-3. Finally, the mean squared error loss function was optimized due to the prediction task being a regression problem.

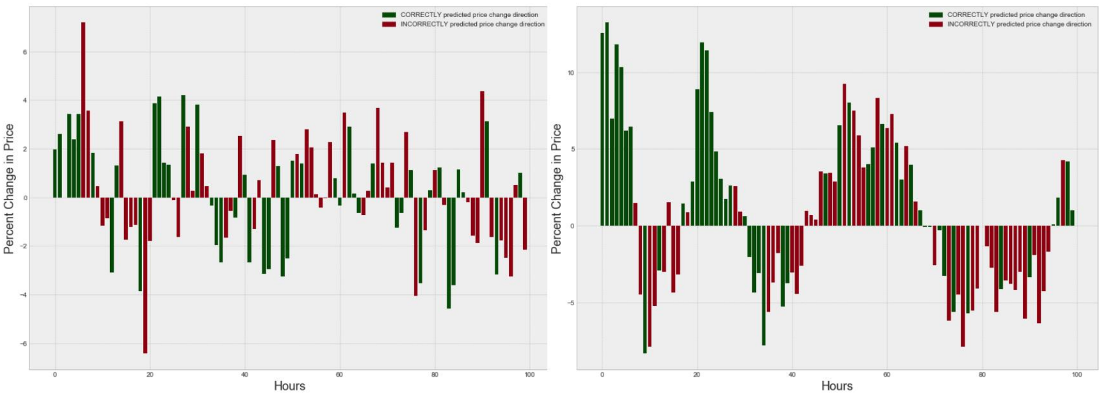
To make overall predictions with the final models, the individual predictions for all data points appearing in the same hour were aggregated, and the average prediction value for each of the five prediction times was calculated. If an average value was positive, a price increase for the time period was predicted; a negative value triggered a prediction of a price decrease.

To evaluate model performance, three different measurements were considered - overall accuracy, accuracy on predictions for which the actual price change was in the largest 20% of all price changes (i.e., largest price increases), and accuracy on predictions for which the actual price change was in the smallest 20% of all price changes (i.e., largest price decreases). Results for each of these measurements are shown in Tables 4, 5, and 6 for each data source and prediction time period . Visual representations of the overall prediction accuracy of the Reddit-based model are shown in Figure 1. Prediction accuracy was fairly consistent across models, so the charts shown are indicative of the headline-based and tweet-based model performances as well.

For each of the three models, prediction accuracy improved as the prediction time increased, with an apparent peak in overall accuracy at a prediction time of 12 hours out. Several factors are at play that may contribute to this trend. First, it naturally takes some time before certain news or announcements manifest as actual changes to the price of a crypto coin. The results show that 12 hours may be the ideal time period for new information to have an effect on coin price. Additionally, the prices of crypto coins are volatile and are affected by a multitude of factors, making price changes over smaller time periods highly randomized and difficult to predict.

Notably, prediction accuracy was higher for all three models on test examples where the actual price change was one of the largest 20% of positive increases than it was when the actual price change was one

Figure 1: Sample Model Performance on Test Set. Bar magnitude represents the actual price change, while bar color represents the correctness of the model prediction.



(a) Reddit Predictions +2 hr in Future

(b) Reddit Predictions +12 hr in Future

of the largest 20% of price decreases. This is likely because prices generally increased over a majority of the training set, so the model was biased towards predicting a price increase when the input signal was ambiguous, resulting in the model mispredicting fewer large increases than large decreases.

Table 4: News Headline Model

Pred. Time	Overall	Largest 20%	Smallest 20%
+1 hr	49.25%	42.65%	51.47%
+2 hr	48.36%	42.03%	45.71%
+6 hr	53.13%	46.48%	48.48%
+12 hr	54.03%	57.97%	44.93%
+24 hr	54.03%	60.00%	46.48%

Table 5: Tweet Model

Pred. Time	Overall	Largest 20%	Smallest 20%
+1 hr	46.77%	46.55%	48.00%
+2 hr	48.12%	50.77%	42.94%
+6 hr	51.73%	57.38%	45.71%
+12 hr	50.83%	57.93%	42.77%
+24 hr	54.89%	60.40%	43.59%

Table 6: Reddit Post Model

Pred. Time	Overall	Largest 20%	Smallest 20%
+1 hr	48.81%	47.97%	51.11%
+2 hr	50.45%	54.76%	47.01%
+6 hr	50.60%	55.74%	50.00%
+12 hr	54.48%	65.29%	46.32%
+24 hr	50.15%	62.50%	36.76%

The most promising results came from the model trained on Reddit posts with upvote scores greater than 10, when predicting the price change 12 hours into the future. This is likely due to the fact that after filtering the text, the Reddit model had by far the most data to train on, combined with the fact that the upvote score filter isolated text that had been deemed interesting/relevant by multiple members of each subreddit.

6 Conclusion and Future Work

This project produced two novel concepts not utilized in previous work: a Word2Vec model that understands semantics and slang utilized within the crypto community, and an RNN-based model that is able to predict price fluctuations for Bitcoin at a rate better than random guessing. The best performing model overall involved a custom trained Word2Vec embedding layer that fed into a bi-directional LSTM trained on Reddit data from relevant crypto communities. While the resultant overall accuracy was only slightly better than random guessing, we believe there is room to further explore. Using these models, it may be possible to predict whether a piece of text will be relevant to a future price change, instead of predicting what that price change will actually be. Additionally, it may be useful to spend time labeling text data based on actual believed sentiment to provide a more easily interpretable signal to the model.

7 Contributions

Connor Lamon

Developed news headline scraping script. Implemented the baseline LSTM model and ran it on the headlines dataset. Trained the custom word2vec model and performed data analysis and visualization on the word vectors to determine the models effectiveness.

Eric Nielsen

Created scripts to obtain coin price data and to label media data. Created script to acquire and label Reddit data. Created and experimented with the baseline Reddit model; also fine-tuned the final Reddit model. Created prediction aggregation and accuracy measurement functions.

Eric Redondo

Developed scripts to acquire, clean and filter, and label both arbitrary Bitcoin/crypto related tweets, and tweets from established Bitcoin developers. Created, experimented with, and analyzed performance of Twitter model and results.

References

[1] Venkata Sasank Pagolu, Kamal Nayan Reddy Challa, Ganapati Panda & Babita Majhi; *Sentiment Analysis of Twitter Data for Predicting Stock Market Movements*; https://arxiv.org/abs/1610.09225; (2016)

[2] Bollen, Johan, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market." *Journal of computational science* 2.1 (2011): 1-8

[3] Evita Stenqvtst & Jacob LÖnnÖ; *Predicting Bitcoin price fluctuation with Twitter sentiment analysis*; http://www.diva-portal.org/smash/get/diva2:1110776/FULLTEXT01.pdf; (2017)

[4] Devavrat Shah & Kang Zhang; *Bayesian regression and Bitcoin*; https://arxiv.org/abs/1410.1231; (2014)

[5] Zhengyao Jiang & Jinjun Liang; *A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem*; https://arxiv.org/abs/1706.10059; (2017)

[6] Bitcoincharts; https://bitcoincharts.com/charts/

[7] Coindesk; https://www.coindesk.com/

[8] Reddit API; https://www.reddit.com/dev/api/

[9] Twitter API; https://developer.twitter.com/en/docs/tweets/search/overview

[10] Cyptocompare API; https://www.cryptocompare.com/api/

[11] Keras; Chollet, Francois and others, https://github.com/keras-team/keras

[12] Tensorflow; https://www.tensorflow.org/

[13] scikit-learn; http://scikit-learn.org/stable/

[14] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems* 2013.

[15] Pre-trained Word2Vec Model; https://code.google.com/archive/p/word2vec/

[16] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. Proceedings of the LREC 2010 Workshop on New Challenges for NLP Framework, pages 45–50, ELRA, http://is.muni.cz/publication/884893/en