Exploring the Influence of News Articles on Bitcoin Price with Machine Learning

Conference Paper · June 2019

CITATIONS
0

READS
106

3 authors, including:

Wenbing Yao
Tsinghua University
2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Exploring the Influence of News Articles on Bitcoin Price with Machine Learning

Wenbing Yao*, Ke Xu†‡, Qi Li§ ¶
*Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
†Department of Computer Science, Tsinghua University, Beijing, China
‡Beijing National Research Center for Information Science and Technology, Beijing, China
§ Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China
ywb16@mails.tsinghua.edu.cn, xuke@tsinghua.edu.cn, qi.li@sz.tsinghua.edu.cn, ¶Corresponding author

***Abstract*—In recent years, cryptocurrencies have become more and more popular around the world, and they are being accepted and used by more countries. Cryptocurrencies are decentralized, and they form an emerging market that is different from stocks. At present, there is already much work around the stock price prediction using news articles, but there are few papers on the cryptocurrency market. In this paper, we aim to research the effects of news articles on bitcoin prices. We extract features from news articles with both commonly used text feature extraction algorithms (e.g., N-Gram and TF-IDF) and SentiGraph, which is a novel text representation method we propose. SentiGraph takes advantages of sentiment analysis and transforms a news article into a graph. Compared with previous feature extraction methods, our experiment results show that this new approach is superior on the prediction accuracy, which also demonstrates the impacts of news articles on the bitcoin price.**

***Index Terms*—Text Mining, Bitcoin Price Trends, Machine Learning**

I. Introduction

Market price forecasting has always been a very challenging task, which appeals to many researchers. At present, there is still no way to predict the price of certain commodities such as stocks accurately. Traditional stock markets have attracted many researchers, while the emerging cryptocurrency markets lack comprehensive research. The cryptocurrency market has developed rapidly in recent years. As of the end of February 2019, the market capacity of cryptocurrencies has exceeded $131.5 billion. Cryptocurrency can be used to purchase goods in some countries and regions, and it can also be bought and sold freely. Bitcoin, as one of the most representative cryptocurrencies, is not controlled by a centralized organization and is maintained by thousands of full-nodes around the world. It can be traded 7x24 hours worldwide, and the market is freer than stocks'.

Many factors can cause price fluctuations in Bitcoin. Some adverse events about Bitcoin, such as the theft of Bitcoin on the Mt.Gox [1] exchange, caused the price of Bitcoin to plummet. As an emerging thing, the price of bitcoin will be affected by the degree of social acceptance. In countries such as Japan and Australia, Bitcoin is considered a legal currency, which leads to a significant increase in the demand for Bitcoin and has a positive effect in the Bitcoin market. Countries around the world are gradually accepting Bitcoin, and this positive impact may last for some time.

Bitcoin is a privacy-preserving currency, and it is different from paying bills with credit cards. Using credit card spending will leave a record of consumption at banks and merchants, but bitcoin is different. Although Bitcoin transactions also leave records on the blockchain, they are anonymous. In the case of using bitcoin properly, it is difficult to match the address of a Bitcoin account to a specific person in real life. Because of this, it is now often used for illegal commodity transactions or criminal activities. Therefore, the price of Bitcoin will also be affected by illegal activities. Bitcoin prices are also influenced by many other factors, such as public opinion, public sentiment, government policies, and the behavior of Bitcoin holders that own a large number of bitcoins.

Daily news contains information about most of the factors that we mentione above and is usually the first-hand data exposing to the public. With large amounts of applications on the mobile phone, people who care about bitcoin transactions can get the latest news and price fluctuations at any time and any place, which could help them decide when to buy and sell.

In this paper, we want to answer the following research problem:

Can the daily news affect the price of bitcoin?

News articles are in text form. However, most machine learning algorithms cannot process raw texts directly. We convert news articles in plain text format into numerical feature vectors that machine learning algorithms can handle. In the field of Natural Language Process (NLP), lots of algorithms and techniques have been proposed to help analyze texts. We explore three traditional text representation methods, which are commonly used in previous works, and we also propose a new approach. The new proposed method is called SentiGraph. It represents a news article as a graph which reserves much more information than traditional methods. The graph is transformed into a dense vector finally for computing simplicity and avoiding over-fitting. With our novel approach, we achieve the best accuracy of 59.0%.

The main contributions of this paper can be summarized as follows:

- To the best of our knowledge, we are the first to explore the relationship between news articles and the Bitcoin price fluctuation.

- We experiment with several machine learning methods using four text representation methods including a novel one we propose.
- Our proposed text representation method, SentiGraph, is a general framework. The new approach is not specific to Bitcoin price prediction task; it could be easily applied to other text analysis tasks.

The rest of the paper is organized as follows: an overview of related literature is summarized in Section II. The experiment framework and our proposed method are described in Section III. The dataset and experimental result are presented the Section IV. We conclude the study in Section V.

II. Related Works

A. Text Representation

One common used method for text representation is term vector model [2]. Terms could be Noun Phasing [3], Named Entity [3], Proper Nouns, etc. Others include N-Gram model [4] and Bag of Words (BOW) [3] model.

Terms based model and BOW lose the order information. Noun Phasing analyzes text with parts of speech and extracts noun phrases. Named Entity approach select terms belonging to certain categories such as locations and organizations. Terms set used in Proper Nouns is a superset of Named Entity and a subset of Noun Phasing. Features of selected terms consist of a vector representing the document. N-Gram model considers a continuous sequence of n words. However, these methods are hard to catch the relationship between words.

B. Bitcoin Price Prediction

Stenqvist et al. [5] collect 2.27 million tweets about Bitcoin from Twitter and extract sentiment information from the data set to predict the price change in the near future. They aggregate the sentiment change during different intervals ranging from 4 minutes to 4 hours, and then they shift these values forward in $1 \sim 4$ times periods to indicate the corresponding price fluctuation. The experimental results show that their best parameters yield an accuracy of 79%. However, their experiments are based on data from just one month, which is not representative.

Akcora et al. [6] extract 400 types of chainlets from Bitcoin blockchain, and then cluster them with Cosine Similarity. They prove the dataset's predictive power with Granger Causality and predict Bitcoin price with random forest model.

Guo et al. [7] predict short-term price fluctuations of Bitcoin with buy and sell orders. They conduct a new generative temporal mixture model that performs better than current models based on time-series and machine learning.

McNally et al. [8] predict the price of bitcoin using historical price data, mining difficulty and hash rate with machine learning method. They achieve the best classification accuracy of 52% and RMSE of 8%.

C. Stock Market Prediction

Bollen et al. [9] attempt to predict the stock market. Seven features are extracted from daily tweets by OpinionFinder and GPOMS. These features are analyzed by Granger causality and then used for training with Self Organizing Fuzzy Neural Networks. In our scenarios, we only use news articles which are different from instant messages. The news is relatively much longer and contains richer information. The length of tweet used in their experiments is limited to 140 while the news' length has no such limitation. Pagolu et al. [10] extract N-gram and $word2vec$ [11] representation for tweet data and use them to train the sentiment classifier. The results from the classifier are used for predicting stock price trend with the logistic regression algorithm and SVM.

Kaya et al. [12] label articles as positive or negative with delta price value in a day and extract $(verb, noun)$ couples that are in the same sentence as features. Feature selection method chi-square is then used to filter out meaningless couples.

Nagar et al. [13] scan the news and filter out *instances* about specific types of stocks. *Instances* could be headlines, sentences, paragraphs, and articles. The number difference between positive and negative words indicates the polarity of an instance. The ratio of the number of positive polarity instances defines the score of a corpus. They find a strong correlation between the corpora scores and the stock price.

Chen et al. [14] use news articles together with other information such as headlines, timestamps, etc. to predict stock prices. In their method, words in sentences are scored to positive (+1, +1.8), and negative (-1, -1.8), the weighted sum of words in a sentence is the sentence score. The scores of all sentences in an article are summed up to give the final score.

Kalyani et al. [15] count the positive and negative words and use the count difference of them as the score of a news article. The TF-IDF method is used for transforming the article into a vector. RF, Naive Bayes, and SVM are used for classification.

III. Methodology

In this section, we first introduce our experimental framework, and then we give a detailed introduction to our proposed text feature extraction method. At last, we provide an example of SentiGraph to make the new approach more clear.

A. Prediction Framework

Our experimental framework is shown in Fig. 1. News articles we collect from the Internet cannot be directly used for the training of machine learning models. They have to be transformed into machine-friendly data formats using text feature extraction methods. The original news data contain a lot of noise information, such as non-English characters, picture information, etc. Before feature extraction, we must remove these useless information through the preprocessor. From the price data, we extract the labels, which indicates the daily changes in the price of bitcoin.

There are some researches related to the prediction of the bitcoin market. However, they analyze the market in a way that is different from that used in our work, in which, we focus on analyzing the influence of news articles on bitcoin's price fluctuation. In previous works, many methods are used to extract features from long texts such as news articles. We

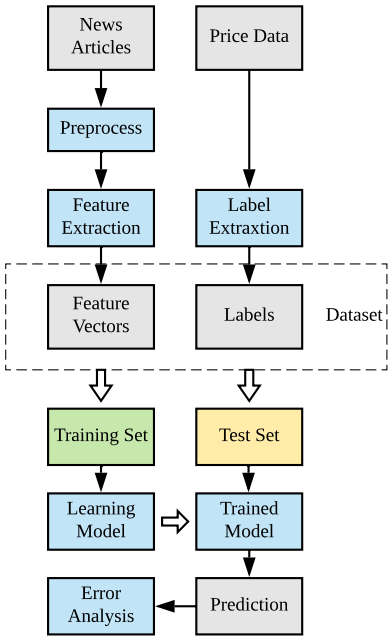

Fig. 1. The framework used in our experiments.

compare our proposed text representation approach with three main text feature extraction techniques, namely, N-Gram, TF-IDF [16], and Doc2vec [17]. These techniques are frequently used in previous works like [10]. We now briefly introduce these techniques:

- *N-Gram.* In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items (such as a word) from a given sample of text. The number N can be chosen as an arbitrary number such as 1, 2, 3. Each N-Gram fragment is assigned a value that indicates its frequency in the whole texts. The values of all the fragments consist of the feature vector of the text.
- *TF-IDF.* Term frequency-inverse document frequency (TF-IDF) is a numerical statistic, intending to reflect how important a word is to a document in a corpus. The TF-IDF is the product of two statistics, term frequency, and inverse document frequency. In the case of the term frequency $tf(t, d)$, the most straightforward choice is to use the raw count of a term in a document. The inverse document frequency is a measure of how much information the word provides, i.e., if it is common or rare across all documents. The TF-IDF value increases proportionally to the times that a word appears and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.
- *Doc2vec.* Doc2vec is an unsupervised algorithm to create a numeric representation of raw texts, regardless of its length. The algorithm is an adaptation of word2vec [18] which can generate vectors for words. The underlying intuition of Doc2vec is that the document representation should be good enough to predict the words in the document. By substituting the input in word2vec architecture from word to sentence, doc2vec can learn a fixed-length vector representing an entire sentence. Doc2vec has been widely used in sentiment analysis.

We design a novel text representation method, i.e., SentiGraph. The new approach is based on the sentiment analysis technique, and the details are introduced in the next subsection.

The dataset is divided into a training set and a test set. We train the machine learning algorithm through the training set, after which we feed the test set to the trained model and then get the predicted result. At last, we analyze the prediction with Error Analysis Module.

B. SentiGraph

In this subsection, we elaborate on the Sentiment Graph (SentiGraph). We explain how to convert a news article to a graph and give an example of SentiGraph generated from an actual news article.

Sentiment analysis is widely researched in recent years. It is useful to help understand the customers' attitudes [19] and predict the stock market [9]. We try to apply this technique to news articles for text mining. We are inspired by the social network in which the behaviors and ideas of one person can influence the others. Intuitively, one entity in an article can have an optimistic or pessimistic attitude towards other ones, and it can impact others positively or negatively. The sentiment graph built from an article records all information about this.

Now we give the formal definition of *SentiGraph*.

Definition. *(SentiGraph)* A sentigraph $SG(V, E)$ is a collection of nodes $V = \{v_1, v_2, ..., v_n\}$ and edges $E = \{(v_i, v_j, sv)|sv$ is the mean of SV, SV is a set of sentiment values of all sentences containing the node pair (v_i, v_j), $1 \leq i, j \leq n, sv \in R\}$. ($R$ represents real numbers.)

Noun pairs (s, d) separated by a verb in one sentence will be added in the graph G. The weight of a pair is the mean of all sentiment values of the sentence containing it. If a sentence contains only one noun, the noun has an edge pointing to itself. The detail is shown in Algorithm 1. Algorithm 1 also returns the occurrence frequency of noun pairs which is helpful when we want to combine graphs.

A simple rule is used to extract noun pairs: Part-of-Speech tagging technique is used to mark up words in texts. Nouns in a sentence is divided into several groups. If there exists a verb between two nouns, these two nouns belong to different groups. Two nouns of a pair are from two adjacent groups.

An implementation of Vader [20] is used in our experiments for sentiment analysis.

C. A SentiGraph Example

News articles contain the latest events and corresponding analysis. In a sentence, a noun which may stand for an entity can express a positive or negative attitude towards another one. We use nodes in a graph to represent nouns and a weighted edge to express the sentiment between two nouns.

A sentigraph generated from an article is shown in Figure 2. The edge of $(bitcoin, mtgox)$ is positive while the sides

Algorithm 1 Build Sentiment Graph

Input: $text$ - news' content
Output: $graph$ - the representation of the input news

function BUILDSENTIGRAPH(text)
  for all $sentence$ **in** $article$ **do**
    $value$ = sentiment_of($sentence$)
    for all $pair$ **in** $sentence$ **do**
      if $pair$ **not in** $graph$ **then**
        $graph[pair].sentiment$ = $emptylist$
        $graph[pair].frequency$ = 0
      end if
      $graph[pair].sentiment$.append($value$)
      $graph[pair].frequency$ += 1
    end for
  end for
  for all $pair$ **in** $graph$ **do**
    $value$ = mean($graph[pair].sentiment$)
    $graph[pair].sentiment$ = $value$
  end for
  return $graph$
end function

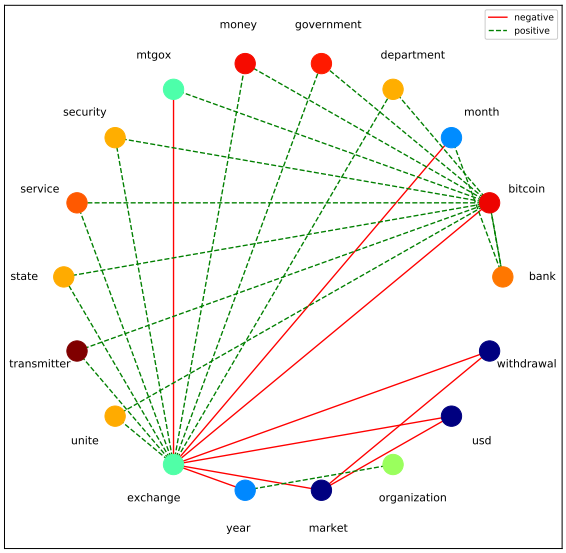

Fig. 2. A sentigraph generated from an actual news article. The color of the node indicates its average sentiment. The lighter the node is, the more positive it is, and vice versa.

of $(exchange, mtgox)$ and $(exchange, bitcoin)$ are negative. This news article talks about the event that large amounts of bitcoins are stolen on MtGox (an online bitcoin exchange). This event hurts the bitcoin market and causes a significant drop in bitcoin price. Intuitively, when the node $mtgox$ comes up with the node $bitcoin$ in some articles, it conveys a negative emotion. So this edge should be analyzed together with the edges of $(exchange, mtgox)$ and $(exchange, bitcoin)$. The graph records all the sentiment information in a news article, and it is more expressive than other methods such as BOW and term-based ones.

IV. Data and Experiments

In this section, we first introduce the data we collect and describe the processing steps, then we present the experiments and the result along with the analysis.

A. Data Processing

1) Data Collection: We crawl news about bitcoin from ten websites, which concentrate on the area of cryptocurrency. All the articles are written in English. We collected 108k+ news articles from March 2012 to February 2019. Each news contains the release date, title, and content. Some news articles that do not give the exact release date are dropped. The JSON file of the data set consumes more than 300MB disk space.

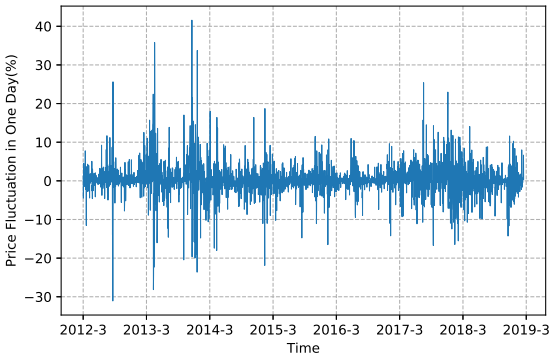

Fig. 3. Bitcoin price data from 2012-3-1 to 2019-2-20.

The price data is collected from $coindesk.com$. It contains the close price of each hour. We extract labels through the daily fluctuation value. The fluctuation value from time t_1 to t_2 is defined in formula (1).

$$fv = \frac{price_{t_2} - price_{t_1}}{price_{t_1}}, \quad (t_1 < t_2) \tag{1}$$

The daily fluctuation values from March 2012 to February 2019 are shown in Fig. 3. From the figure, we can see that these fluctuations are similar to randomly generated numbers, which indicates that it is challenging to predict the trends. The label is defined according to the sign of fluctuation value as shown in formula (2).

$$label = \begin{cases} rise, & \text{if } fv > 0 \\ drop, & \text{if } fv \leq 0 \end{cases} \tag{2}$$

We use the news in one day to predict the price fluctuation in that day. It means that in formula (1), t_1 is 0 o'clock and t_2 is 24 o'clock in the same day.

2) Feature Extraction: In the methods of N-Gram, TF-IDF, and Doc2vec, we use open source packages to extract features. In the method of SentiGraph, we make an implementation of SentiGraph with $python$ programming language. The $NLTK$ [21] package is used in our experiments. The function pos_tag in the $NLTK$ package is used for helping to extract nouns in a sentence and the $vader$ module in $NLTK$ is used

for sentiment analysis. For the reason that our purpose is to predict the price fluctuation in one day, we combine the graphs into a large graph. First, we extract one graph for each article, and after obtaining all the sentigraphs of the news articles, we combine them according to the date. Sentigraphs $\{G_1, G_2, ..., G_n\}$ of news articles belonging to the same day are combined.

$$w(s,d) = \frac{1}{n}\sum_{i=1}^{n} G_i[(s,d)] * F_i[(s,d)], \tag{3}$$

F is the pair frequency dictionary.

New weight of an edge (s, d) in G is calculated with the formula (3). Note that, we use weighted mean to combine

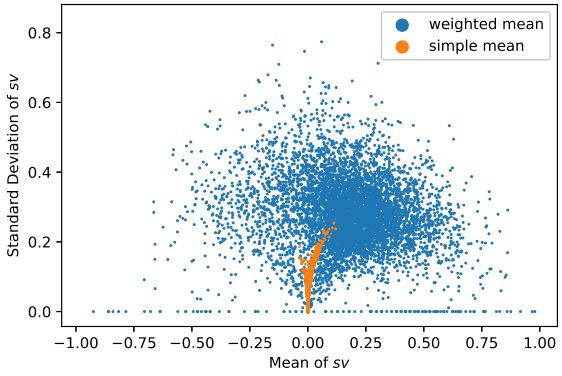

Fig. 4. Each point in the figure represents a noun. The mean and standard deviation value are calculated from the sv values of all pairs containing the noun.

two graphs instead of just calculating the mean value with $F_i[(s,d)] \equiv 1$ in the formula (3). The reason is that values with weighted mean contain more distinguishable information than the more straightforward method without considering the frequency. Figure 4 shows the mean and standard deviation of the sentiment values with and without pair weights. From Fig. 4, we can see that if we do not consider the pair frequency, then the mean of sentiment values is concentrated around 0, and most of the standard deviation values are between 0 and 0.2. However, if we use weighted summation, these two values will increase significantly, which dramatically increases the degree of discrimination.

3) Dimension Reduction: In our SentiGraph method, the number of noun pairs extracted from the whole news articles is more than three million. The occurrence times distribution is shown in Figure 5. About half of the pairs appear less than ten times. It is evident that if the frequency value of a pair is too small, it is not able to contribute to the prediction task too much. We define a threshold T_{occur} and filter out pairs the number of which is under T_{occur}. The best value of T_{occur} may differ among different tasks. In the experiments, we found fifty is a proper value, and the results we show in the following is based on this T_{occur} value. A graph is converted to a sparse vector in which every position is for a specific pair. The vector representation for a graph can be handled by machine learning algorithms easily. We assign an index value for each pair. If a pair is in a graph, its sentiment value is set in the corresponding position; otherwise, it is set to zero.

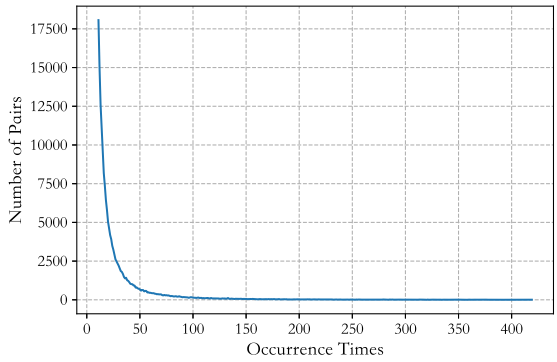

Fig. 5. Distribution of Occurrence Times.

To accelerate training and ease the problem of over-fitting, we convert sparse vectors of the three methods, namely, N-Gram, TF-IDF, Doc2vec, into dense vectors of low dimension with Principal Component Analysis (PCA) algorithm.

B. Experimental Result

TABLE I
THE RESULTS OF DIFFERENT FEATURE EXTRACTION METHODS WITH THE FOUR MACHINE LEARNING ALGORITHMS.

FEM	Model	Accuracy	Precision	Recall	F1
N-Gram	SVM	0.560	0.317	0.560	0.404
	RF	0.523	0.516	0.523	0.518
	MLP	0.513	0.489	0.513	0.487
	Logistic	0.560	0.550	0.560	0.548
TF-IDF	SVM	0.530	0.493	0.530	0.478
	RF	0.543	0.533	0.543	0.534
	MLP	0.513	0.484	0.513	0.480
	Logistic	0.527	0.526	0.527	0.526
Doc2vec	SVM	0.547	0.539	0.547	0.541
	RF	0.533	0.524	0.533	0.525
	MLP	0.540	0.544	0.540	0.541
	Logistic	0.550	0.547	0.550	0.548
SentiGraph	SVM	0.557	**0.691**	0.557	0.420
	RF	0.557	0.550	0.557	0.538
	MLP	0.587	0.590	0.587	0.557
	Logistic	**0.590**	0.589	**0.590**	**0.571**

In our experiments, the data on the latest 300 days is used for the test, and the rest data is used for training. Table I shows the results. The precision, recall, and f1 values are the weighted averages of the $rise$ and $drop$ classes. We compare the performance with four machine learning algorithms, namely, Support Vector Machine (SVM), Random Forest (RF), Multilayer Perceptron (MLP), Logistic. During our experiments, we find that SVM models are more difficult to train compared with the other three machine learning

algorithms. It is easy for the model to present the results in which all the predictions are $rise$. This phenomenon may result from the imbalance of the dataset, in which there are slightly more $rise$ data than $drop$ data. The basic information of our dataset is given in Table II.

TABLE II
BASIC INFORMATION OF OUR DATASET.

# of News	# of samples	# of $rise$ samples (percent)	# of $drop$ samples (percent)
108433	2224	1017 (45.73%)	1207 (54.27%)

The results show that the prediction accuracy of different feature extraction methods is higher than 50%. However, no means generate an accuracy higher than 60%, which indicates the limitations of the news articles. Our proposed method achieves the highest classification accuracy of 59% with Logistic algorithm. Our method also generates the best recall of 0.59 and the best F1 of 0.571. The best precision appeals in the experiments with our proposed SengiGraph using SVM. The results also indicate that the average performance of Sentigraph among the four machine learning algorithms is superior to the other three feature extraction methods. Logistic shows better performance than the other three machine learning algorithms except with TF-IDF. Averagely, TF-IDF performs worst in the classification task. We also implement some methods used in previous works such as [12], and find the accuracy of their methods are under 55%, which is worse than our method.

V. CONCLUSION

In this work, we study the influence of daily news on the price of Bitcoin. We compared the performance of four feature extraction methods by training four machine learning models. Notably, we propose a novel text representation method, i.e., SentiGraph, which achieves the highest accuracy of 59%. Moreover, our method's performance is better than traditional N-Gram, TF-IDF, and Doc2vec methods on the other three metrics. The experiment results demonstrate that news articles do have impacts on the price of bitcoin. Compared with previous work, our approach successfully utilizes this kind of impacts and improve prediction accuracy.

ACKNOWLEDGMENT

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB0803405, the National Natural Foundation of China under Grants 61825204, 61572278, and U1736209, and Beijing Outstanding Young Scientist Project.

REFERENCES

[1] A. Cheung, E. Roca, and J.-J. Su, "Crypto-currency bubbles: an application of the phillips–shi–yu (2013) methodology on mt. gox bitcoin prices," *Applied Economics*, vol. 47, no. 23, pp. 2348–2358, 2015.

[2] G. Giannakopoulos, P. Mavridi, G. Paliouras, G. Papadakis, and K. Tserpes, "Representation models for text classification: a comparative analysis over three web document types," in *Proceedings of the 2nd international conference on web intelligence, mining and semantics*. ACM, 2012, p. 13.

[3] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The azfin text system," *ACM Transactions on Information Systems (TOIS)*, vol. 27, no. 2, p. 12, 2009.

[4] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos, "Words versus character n-grams for anti-spam filtering," *International Journal on Artificial Intelligence Tools*, vol. 16, no. 06, pp. 1047–1067, 2007.

[5] E. Stenqvist and J. Lönnö, "Predicting bitcoin price fluctuation with twitter sentiment analysis," 2017.

[6] C. G. Akcora, A. K. Dey, Y. R. Gel, and M. Kantarcioglu, "Forecasting bitcoin price with graph chainlets," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 765–776.

[7] T. Guo and N. Antulov-Fantulin, "Predicting short-term bitcoin price fluctuations from buy and sell orders," *arXiv preprint arXiv:1802.04065*, 2018.

[8] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, 2018, pp. 339–343.

[9] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *CoRR*, vol. abs/1010.3003, 2010. [Online]. Available: http://arxiv.org/abs/1010.3003

[10] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of twitter data for predicting stock market movements," in *International Conference on Signal Processing, Communication, Power and Embedded System*, 2017, pp. 1345–1350.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[12] M. Y. Kaya and M. E. Karsligil, "Stock price prediction using financial news articles," in *2010 2nd IEEE International Conference on Information and Financial Engineering*. IEEE, 2010, pp. 478–482.

[13] A. Nagar and M. Hahsler, "Using text and data mining techniques to extract stock market sentiment from live news streams," in *International Conference on Computer Technology and Science (ICCTS 2012), IACSIT Press, Singapore*, 2012.

[14] J. Chen, A. Chai, M. Goel, D. Lieu, F. Mohamed, D. Nahm, and B. Wu, "Predicting stock prices from news articles," *The Undergraduate Statistics Association-Project Committee journal. APPENDICES*, 2015.

[15] J. Kalyani, H. N. Bharathi, Prof., and R. Jyothi, Prof., "Stock trend prediction using news sentiment analysis," *ArXiv e-prints*, Jul. 2016.

[16] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of tf* idf, lsi and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758–2765, 2011.

[17] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[18] T. Mikolov, I. Sutskever, C. Kai, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.

[19] G. Hu, P. Bhargava, S. Fuhrmann, S. Ellinger, and N. Spasojevic, "Analyzing users' sentiment towards popular consumer industries and brands on twitter," *arXiv preprint arXiv:1709.07434*, 2017.

[20] C. H. E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*, 2014.

[21] J. Perkins, *Python 3 text processing with NLTK 3 cookbook*. Packt Publishing Ltd, 2014.