Twitter posts as indicator for future price of Bitcoin

by

Temirlan Ulugbek uulu

Bachelor Thesis in Computer Science

Prof. Michael Sedlmair
Bachelor Thesis Supervisor

Date of Submission: May 12, 2018

Jacobs University — Computer Science

With my signature, I certify that this thesis has been written by me using only the indicates resources and materials. Where I have presented data and results, the data and results are complete, genuine, and have been obtained by me unless otherwise acknowledged; where my results derive from computer programs, these computer programs have been written by me unless otherwise acknowledged. I further confirm that this thesis has not been submitted, either in part or as a whole, for any other academic degree at this or another institution.

Signature   Place, Date

Abstract

Consider this a separate document, although it is submitted together with the rest. The abstract aims at another audience than the rest of the proposal. It is directed at the final decision maker or generalist, who typically is not an expert at all in your field, but more a manager kind of person. Thus, don't go into any technical description in the abstract, but use it to motivate the work and to highlight the importance of your project.

(target size: 15-20 lines)

Contents

1 Introduction

1.1 Goal and Motivation

Since early ages of financial markets, people have been trying to predict the future price of traded assets. With knowledge of future price or at least the direction of the price movement, one can gain profits or even more - manipulate the market. Knowing the price for the coming year, month, day or hour gives you by far more opportunities than simple buy low and sell high. This way or the other, knowing the future price is always good. This paper describes another attempt to predict future price of an asset. However, in this paper we dont try to predict the future price of ordinary financial market with usual stocks or physical commodities. We deal with cryptocurrency, in particular Bitcoin. To understand the core idea of prediction, one has to understand what is cryptocurrency and how it is traded.

1.2 Understanding Bitcoin

Initially, Bitcoin was invented as a currency for peer-to-peer electronic cash system, where users dont have to trust anybody. One can think of bitcoin as electronic money. All the transactions and creations of this coin are stored in the blockchain data-structure. The idea of blockchain was proposed nearly a decade ago in 2009 by someone who named himself as Satoshi Nakamoto [3]. There are many good sources out there that try to explain the technology behind this currency, but that knowledge is not necessary for the understanding of this paper. Here are few things that we need to understand about bitcoin.

Satoshi Nakamoto wrote in his paper: We have proposed a system for electronic transactions without relying on trust [3]. And by that he meant that one doesnt have to trust anyone to manage his bitcoins. No bank, government or other third party in between. As long as you keep your private keys safe, no one will be able to steal your bitcoins. That is achieved by a huge computational power needed to maintain the bitcoin blockchain. Huge amount of computers all over the world contribute to this network, i.e. they do computations to verify transactions and if one successfully finds the solution, he/she gets some amount of bitcoin for his work. At the moment of writing this, the total computational power is around 30 000 000 tera hashes per second [4]. If someone wants to do illegal transactions or steal someones bitcoins, he has to have the majority of the CPU power, which is nearly impossible to achieve. The difficulty of verifying transaction brings us to one of the major problems of this blockchain system. Transactions are costly and slow. Average transactions take couple of hours and may be faster if you are willing to pay more [5]. And there is no guarantee that your transaction will be confirmed at all. Theoretically it may be stuck in the pool of transactions forever. This makes it a bit inconvenient for bitcoin to be the substitution for the current payment ways. However, the great technology and the idea of trustless payment system made bitcoin a great asset for investments. Bitcoin became a digital gold and is currently traded in tens, if not hundreds, of exchanges all over the world. Anyone who has some money and is willing to invest can do so. Over the past years, there was tremendous growth of bitcoin price [6]. This growth leads to increase of people investing, which in its turn leads back to price growth.

With time, thousands alternatives of bitcoin were created [7]. Mostly they have the same idea of blockchain and decentralization, but have their own features and advantages. Those alternative coins are shortly called alt-coins. Even though bitcoin is not the only cryptocurrency out there, we will analyze only bitcoin price in this paper, since it was the first one of a kind and is still leading among all altcoins according to the market capitalization [7].

1.3 Difference between crypto-market and other financial markets

From a first glance, cryptomarket can seem nothing else but another type of financial market. But they are completely different.The major difference is that there is no real physical commodity or some company stock in cryptomarket. Bits in the agreed blockchain is all what is traded. The price is purely on a level of supply and demand. The highest price that the people are willing to pay and the lowest price for which others are willing to sell. No real world factors affect the price, except the ones that affect the mood of traders. Some people might believe that the price of bitcoin is controlled by people or group of people with a lot of money, also called ”whales”. But mostly whales are asleep, or in other words not actively trading in the markets. No doubt that they can change the direction of the market completely if they want to. But mostly, the price depends on the smaller active traders, at least that is what we assume here. We make a claim that bitcoin price depends on the mood of people all over the world. Thus, if this claim is true and we have the mood of all bitcoin traders in numbers, we could predict the future price.

1.4 Twitter as source of emotional state

Now, to see if the claim in previous section is true, we need a way to extract the emotional state of people in digital format in order to be able to perform computations and predict the future price. And here we involve the Twitter. We make another claim and assume that Twitter has the emotional state of people trading the Bitcoin. We extract all the Twitter posts related to bitcoin and perform a linear regression on them together with history price to find the best predicting function. After getting the predicting function, we can apply it on the current state of peoples mood all over the world and get the future prices for bitcoin.

In order to check the above claims, we extract all the bitcoin related tweets and do sentiment analysis on them. VADER [8] was used in order to extract the polarity of Twitter posts, to be exact, it determined how negative or positive they are. Those sentiment scores and the historical bitcoin price data then have been fed into linear regression algorithm and the optimal predicting function have been obtained.

However Twitter doesn’t allow to extract tweets older than on week and we don’t have the full flexibility here. Even though if the algorithm was fed only with one month of data, the results were pretty impressive. The weekly predictions using the Twitter had accuracy between 66-100%. Provided results should strongly support the claims made earlier and encourage people to investigate more in the direction of considering the Twitter as one of the significant indicators for the price predictions of cryptocurrencies.

2 Related Work

2.1 Unrelated prediction algorithms

One has to understand the difference between previous works that have the same goal - predicting the future market state. Many good algorithms were developed in predicting the future price in other financial markets. Despite the fact that some of them achieved very good results, those works have very little to do with the described in this paper. That is mostly because the markets are completely different as stated in section 1.3. Thus, we only should look at the works related to the cryptocurrency market. Even if the field is relatively new, a lot of good attempts were done in developing predicting algorithms. Different algorithms use different input data. In most of the cases, in cryptocurrency markets, people try to analyze the historical price of the coin. We will not be looking at those kind of works and will focus only the ones that are very similar to what we tried to do here.

2.2 Closely related work

2.2.1 Predicting Bitcoin price fluctuations with Twitter sentiment analysis [1]

As the paper states itself, the work is ... done by a naive method of solely attributing rise or fall based on the severity of aggregated Twitter sentiment change. The work tries a very naive approach and sees if the increase of aggregated Twitter sentiment leads to some direct changes in price. As a result of the work, they get fluctuations of the price in the future. [1] Knowing the fluctuations of the price is nearly useless to the traders, unless the accuracy is 100%. Imagine the case if the output of the algorithm for the next three hours is [decrease, increase, increase]. This might be the case for [-10%, +1%, +1%] as well. Thus trader wont be able to get the most profit, unlike the case if he knew the exact magnitude of the price movement.

2.2.2 Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis [2]

This work does some more advanced work in terms of data processing. It performs multiple ML techniques on the Twitter and Bitcoin data that has been collected in around 20 days [2]. First of all, when comparing to the big history that Bitcoin already has, 20 days of data is a very small dataset to train your system. Moreover, the data has been collected from November 15th to December 3rd in 2015. As Bitcoin usually close to end of the year, the price was very unstable during that period. Somewhat steady down-trend of the price abruptly turned into fast growth. This might be not the very best period to collect the data, but we can only do assumptions here since their work hasnt been tested on other data than the one collected during that period. The main goal described in that paper was to again get the binary result or, to be more precise, to know if the price will increase or decrease, which, as we discussed above, is not very useful. Two approaches were made to achieve the result: feeding words from tweets separately into the learning algorithms and feeding sentiment scores of the tweets into the algorithms. We dont cover the first approach, since it is not closely related to the our work. However, the second approach they tried come very close to what has been done here. They took the whole tweet, did sentiment analysis on it and used those labels as input for Naive Bayes. The main flaw here was that they used text-processing.com as their sentiment analizer, which would calculate the positivity, negativity, and neutrality scores [2]. As has been stated in the web-site itself, the sentiment analyzer is composed of 2 classifiers trained on movie reviews. If your text is not similar to movie reviews, then its less likely to make a correct guess [9].

Unlike the papers described above, we try to predict the exact price in the future instead of predicting just the binary increase/decrease of the price. This information will be much more useful for traders. Moreover, we use linear regression to get the optimal predicting function instead of naive approach. As an input to our linear regression, we use polarity scores of tweets. We use VADER to do sentiment analysis on Twitter posts and will later see how VADER outperforms other algorithms. Data for one month has been collected, which is still relatively small dataset, but both price downtrend and uptrend were observed during the period of data collection.

3 Design and Implementation

Very general idea about the design of this work is this: we take all the bitcoin related posts from Twitter, do sentiment analysis on them using Vader, feed this sentiment scores into linear regression, get the optimal predicting function, and finally apply it to the current state to get the future prices.

3.1 Data collection

Data collection is hard when you are dealing with Twitter, because one cant extract old tweets using the official Twitter API. The Twitter Search API searches against a sampling of recent Tweets published in the past 7 days [10]. One has to pay to get more privileges. Using some third party APIs, like GetOldTweets-python [11], wouldnt do the job as well, because they use the Standard Search API in that back. ... its important to know that the standard search API is focused on relevance and not completeness. [10]. Thus, using third party APIs cant get you complete data as has been seen during the process of data collection. Using GetOldTweets [11], we tried to get Bitcoin related tweets since the beginning of 2018, but couldnt get a satisfactory results. It returned only around 1 million tweets, which cant be a complete dataset by any means, given the fact that we were getting more than 100 000 bitcoin related tweets per day when getting the live streaming. Thus, obtaining the complete dataset of old Twitter posts was not possible. Instead, tweets were collected using live stream. Tweepy [12] was used to interact with Twitter API. It is python library, which gets the job done. Every tweet, which contained btc/bitcoin in it, was streamed and stored. In order to be able to extract tweets 24/7, the web-server was rented from Aruba Cloud [13] with minimal requirements: 1GB RAM, 20GB space. MySQL database was set-up on the same server to store the data. The data collection has started on March 30th and around 5 million Twitter posts were stored in approximately 1 month.

3.2 Sentiment analysis

What we wanted from tweets is to know how positive or negative they are. Assuming that we have very huge stream of tweets, we need this analysis to be very quick and preferably as accurate as possible. VADER (Valence Aware Dictionary for sEntiment Reasoning) [8] is the perfect tool we could use for this. It is very computationally cheap and accurate at the same time [8]. Moreover, in the latest update, they made a major improvement in performance: Reconstructing for much improved speed/performance, reducing the time complexity from something like O(N$\hat{4}$) to O(N) [14]. With hundreds of tweets arriving to our server every day, performance was very important when making decision in which tool to use. ... a corpus that takes a fraction of a second to analyze with VADER can take hours when using more complex models like SVM (if training is required) or tens of minutes if the model has been previously trained. [8] Other than performance advantages, it has transparency. VADER is an open-source tool that can easily be inspected and extended [8] [14]. This is important if we would like to introduce it a cryptocurrency related lexicon. VADER also doesnt require any training dataset and is completely self-contained [8]. Given all this conveniences, this tool still outperforms other well-known tools and techniques. When looking at social media context VADER outperforms seven well-established sentiment analysis lexicons (LIWC, GI, ANEW, SWN, SCN, WSD, and Hu-Liu04 opinion lexicon) and four machine learning algorithms (Naive Bayes, Maximum Entropy, SVM-Classification, and SVM-Regression) [8]. Thus, by considering all mentioned factors, VADER was chosen to do the sentiment analysis for this work.

3.3 Feature vectors

In order to be able to use linear regression, we need to think of what should the input and output vectors should contain to meet our problem well. Linear regression was performed on different input data or to be more precise, three types of input: historical price, tweets vector, and combination of these two. Output of our predicting algorithm will be the price vector. Predictions in this work are done for different frame-widths and intervals. This terminology was chosen for this paper with the following definitions. *Frame-width* is the length of the period of time for which the prediction is being made. For example if the frame-width is 1 hour, we are trying to predict the price over the next coming hour. *Interval* is the distance between predictions. For example if the frame-width is 1 hour and the interval is 1 minute, we are trying to predict the price for each minute of the coming hour, or in other words, we do 60 predictions for the coming hour. The terms frame-width and interval will be now used without further explanations.

3.3.1 Price vectors

Price vector are the vectors representing price of Bitcoin over certain period in the timeline or in our case the whole frame-width. In all cases of linear regression done in this work, the output vector is the price vector for the future. In some cases, price vectors of the past are used as input vectors as well. The structure of the price vector is straightforward. If we have frame-width of 1 day and interval of 1 hour, our price vector would be of dimension 24 or in other words, it would contain hourly price information for the whole 24 hours or 1 complete day. Here, we make a small change. Instead of storing simply the prices, we store the price change in percentage from the beginning of the frame. Thus, the first value of the price vector would always be zero since that is the point of time which we take as our baseline. The next value of the vector would be the price change after one interval. If we continue our example, and the price changed from 100 to 110 in one interval, first two values of the price vector would be 0 and 10. This way we ensure that we are looking at the trend instead of looking at the concrete price changes and being bound to some time periods. The price vector from 2010 can now be similar to 2018 and our prediction algorithm would recognize it better. Imagine if we had prices instead of price changes. Then, the price vector from 2010 would contain couple of dollars, if not cents, unlike the vector from 2018 which would contain thousands of dollars. Using our version of price vectors, we reduce the size of vector by 1, since the first value will always be zero. That place can be reserved for future use. The very straight forward way to use it would be to store the initial price as the first value of the vector.

3.3.2 Tweet vectors

When talking about hundreds of thousands of tweets, extracting the feature vector from it is not that obvious. Keeping in mind the idea/claim, that the price depends on the mood of people all around the world, all we need is a feature vector that, simply said, can tell how happy or sad is Twitter Bitcoin community. The raw dataset we have after processing tweets with VADER is list of values (or sentiment scores of tweets) ranging between -1 and 1, where -1 is completely negative and 1 is the opposite. Now we want to divide them into M partitions and calculate the percentage of tweets coming to each partition. For example if M would be 2, we would just have two partitions: [-1,0), [0,1]. Or in other words, we just calculate what part of tweets is negative and what part is positive. Thus, we our tweet vector would have dimension of 2. Here is what was described shortly:

$$Mpartitions \Rightarrow VTweet \in \mathbb{R}^M \tag{1}$$

$$VTweet_i \in [0,1], where\ i = 1, ..., M \tag{2}$$

$$\sum_{1}^{M} VTweet_i = 1 \tag{3}$$

However, it might be tricky to choose the optimal M Its not obvious what resolution would be enough to preserve enough data. Moreover, the value of optimal M may depend on the frame-width and interval. For example if we are trying to predict the price for the coming year, just the small M may be sufficient to find out the general mood for long term predictions. Thus, the testing data was divided into parts and cross validation has been performed for each frame-width and interval to find the optimal M. The error was computed for different values for M starting with 2 and the error was calculated on each iteration. M was being increased until the testing error stopped decreasing or the decrease was below certain threshold. The optimal Ms that are discovered by doing cross-validation are shown in the Results section of this paper.

3.4 Performing linear regression

Linear regression is not the most advanced machine learning technique and most people might say that other techniques, like neural networks, are much better at approximating functions or doing price predictions. However, linear regression is the perfect starting point. If the correlation between the mood of Twitter community and Bitcoin price is defined by some non-linear function, linear regression would most probably fail at making predictions. But due to its simplicity and computational cheapness, linear regression was chosen as a starting point. If mood can be mapped to price by some linear affine mapping we would already get good predictions and there would be no need to do computationally expensive and advanced techniques. Even if predictions wont be as good as we wish they were, linear regression could make the further direction of research clearer and getting even somewhat good (e.g. better than random) could motivate people to investigate more into this field.

As mentioned above, we have three types of inputs to feed into the linear regression: price vectors, tweet vectors, and combined price-tweet vectors. In all three cases, we perform standard linear regression procedure. We divide our data instances into training and testing (validation) sets. We learn the optimal function or optimal matrix from the set of training instances and we evaluate the error of this optimal function by applying it on the validation set. By doing this on different datasets, or to be more precise on only prices, only tweets, and price-tweets combined, we could learn how tweets could be useful when predicting future price using linear regression.

3.5 Result evaluation

To evaluate the results we calculate the *mean deviation* for each prediction. For example if we have our prediction $VPrices_{1h-1d}$ for the next day with predictions for every hour, we calculate by how much did it deviate from the true value. Since we are already computing the price change instead of prices, calculating the mean deviation would give us the average percentage by which we were away from the actual value. If we express it in expression, the following formula is used to compute the mean deviation of the prediction:

$$\Delta = \frac{1}{||VPrice||} \sum_{i=1}^{||VPrice||} |VPriceTrue_i - VPricePredicted_i| \tag{4}$$

The reason we compute mean deviation is that it has the real world meaning and is more useful for traders than, for example, mean square error.

The second value used for evaluation is side accuracy or simply *accuracy* from now on. The accuracy is the percentage of correctly predicted sides (positive or negative price change). The same evaluation value is used in the papers mentioned earlier [1] [2]. The following formula is used to compute the accuracy of the prediction:

$$Accuracy = \frac{\sum TruePositive + \sum TrueNegative}{\sum AllPredictions} \tag{5}$$

Accuracy might be especially useful in cases when the deviation is small. For example with frame-width being 1h, the mean deviation will not be that big because price does not change much in 1 hour. In this case mean deviation stays small even if we predict the constant price. Accuracy helps us out in such cases and allows to get more insight of how good is the prediction.

Another reason of calculating the accuracy of predictions is to be able to compare the results to the results achieved in the earlier works [1] [2]. Since both papers used accuracy to evaluate their work, we can easily compare the results achieved.

4 Evaluation

Here is the whole process step-by-step to evaluate how good the predictions using linear regression are:

1. **Set the frame-width and interval:** Frame-width determines how far do we want to do the predictions and interval determines the distance between individual predictions within that frame.

2. **Construct the input and output feature vectors:** Linear regression needs pairs (x,y) to learn the optimal function and to later evaluate how good is it. Depending on what we are feeding in, we need to construct price vectors, tweets vectors or both. The input x will be the respective data for the past time-period of length frame-width. And the output y will be the price informations for the coming time period of length frame-width. So, if we have frame-width of 1 day and interval of 1 hour, our input vector would contain price and/or tweet information for the last day and the output information would contain the price information for the coming day with interval of 1 hour. And every day would be one data instance.

3. **Divide the dataset:** Now that we have a dataset of input and output pairs, we can compute and evaluate the optimal predicting function. For that we need to divide our dataset into training and validation sets. This part is actually done in iterations in order to find the average mean deviation (cross-validation). For example if we have three pairs, first we take 1,2 as training instances, learn predicting function from it, check the mean deviation and accuracy by testing the function on 3. Then we use 2 as validation and 1,3 as training set. We again compute the error and now use 1 as validation set. After all partitions of the set were used as validation set, we compute the average deviation and accuracy. This way we try to avoid bad training and testing data separation. According python libraries were used to do the dataset separation [15].

4. **Compute optimal predicting function:** To compute the optimal predicting function, the comprehensive algorithm from the Machine Learning lecture notes was followed. Here is the algorithm described in the notes [16] with minor modifications:

Data: A training set $(x_i, y_i)_{i=1,\dots,N}$ of m-dimensional input vectors x_i and k-dimensional target vectors y_i
Result: A $k \times m$ dimensional weight matrix W_{opt} which solves the following linear regression objective:

$$W_{opt} = \underset{W \in \mathbb{R}^{k \times m}}{\operatorname{argmin}} \sum_{i=1}^{N} ||Wx_i - y_i||^2 \tag{6}$$

Step 1: Sort the feature vectors row-wise into an $N \times m$ matrix Φ and the targets into an $N \times k$ matrix Z.
Step 2: Compute the result by $W'_{opt} = (\Phi'\Phi)^{-1}\Phi' Z$
Step 2*: We perform slight modification in this step to avoid numerical issues like inverting the singular matrix. These issues can be avoided by taking pseudo-inverse [16]:

$$W'_{opt} = pseudoInverse(\Phi)Z \tag{7}$$

In python we could implement it the following way:

$$Wopt = (np.linalg.pinv(Phi).dot(Z)).T \tag{8}$$

where matrices are 2D numpy [17] arrays. ”This still gives optimal weights” [16].

5. **Evaluate the optimal predicting function:** Now we that we have our optimal predicting function (more precisely, optimal matrix W_{opt}), we can evaluate it on validation set. For each predicted value $W_{opt}x_{test}$ we calculate the mean deviation and in the end, we look at the average of all calculated mean deviations. We also calculate the average accuracy for all predictions.

4.1 Results

In this discussion, we will discuss the results achieved using the techniques explained earlier. As have already been mentioned above, we have three ways of predicting the price using linear regression or to be more precise, we have three different inputs into our algorithm. In all three cases our goal is to predict future price, thus output will always be a price vector (3.3.1). First we try to predict the price using only the old price data, thus the input will be the price vector. Then we add the sentiment data of Twitter posts, that is we append tweet vector (3.3.2) to the price vector and see how does our predictions change by analyzing the error of predicting function. And thirdly, we use only Twitter data, thus input instances will be tweet vectors only. Throughout the process, we evaluate the performance by computing mean deviation (4) and accuracy (5) for our predictions. In order to avoid by (un)luckily choosing (not)good training and testing data, we do cross validation and show the average mean deviation and accuracy. In our case, we took 4 iterations (75/25 split). Here is the pseudo-code of how this was done:

Algorithm 1 Fair prediction evaluation

```
Require: $\Phi \leftarrow$ All Inputs, $Z \leftarrow$ All Outputs
Ensure: $Z_i$ is corresponding output of $\Phi_i$
  $kf \leftarrow KFold(n\_splits = 4)$ [15]
  $meanDeviations \leftarrow [\,]$
  $accuracies \leftarrow [\,]$
  for $train\_index,\ test\_index$ in $kf.split(\Phi)$ do
    $\Phi_{train} \leftarrow \Phi[train\_index]$
    $Z_{train} \leftarrow Z[train\_index]$
    $\Phi_{test} \leftarrow \Phi[test\_index]$
    $Z_{test} \leftarrow Z[test\_index]$
    $W_{opt} \leftarrow getOptimalPredicting(\Phi_{train}, Z_{train})$
    $dev,\ acc \leftarrow evaluateOptimal(W_{opt}, \Phi_{test}, Z_{test})$
    $meanDeviations.\mathbf{append}(dev)$
    $accuracies.\mathbf{append}(acc)$
  end for
  $meanDeviation \leftarrow \mathbf{average}(meanDeviations)$
  $accuracy \leftarrow \mathbf{average}(accuracies)$
```

4.1.1 Using price data only

4.1.2 Using price and Twitter data

First of all, before we can proceed to computing and evaluating the optimal predicting function, we need to find the optimal number of partitions (1) in the tweet vector.

4.1.3 Using Twitter data only

4.2 Discussion

5 Conclusions

Summarize the main aspects and results of the research project. Provide an answer to the research questions stated earlier.

(target size: 1/2 page)

6 Future Work

References

[1] Jacob Loennoe Evita Stenqvist. Predicting bitcoin price fluctuation with twitter sentiment analysis, June 2017.

[2] Stephanie Rosales Stuart Colianni and Michael Signorotti. Algorithmic trading of cryptocurrency based on twitter sentiment analysis.

[3] Satoshi Nakomoto. Bitcoin: A Peer-to-Peer Electronic Cash System.

[4] Blockchain.info. https://blockchain.info/charts/hash-rate, May 2018.

[5] Blockchchain.info. https://blockchain.info/charts/avg-confirmation-time, May 2018.

[6] Coinmarketcap. https://coinmarketcap.com/currencies/bitcoin/, May 2018.

[7] Coinmarketcap. https://coinmarketcap.com/, May 2018.

[8] Eric Gilbert C.J.Hutto. Vader: A parsimonious rule-based model for sentiment analysis of social media text, 2014.

[9] Text-Processing.com FAQ. http://text-processing.com/docs/faq.html, May 2018.

[10] Official Twitter API. https://developer.twitter.com/en/docs/tweets/search/overview/standard, May 2018.

[11] Jefferson-Henrique. https://github.com/jefferson-henrique/getoldtweets-python, April 2018.

[12] Tweepy. http://www.tweepy.org/, May 2018.

[13] Aruba Cloud. https://www.arubacloud.com/, May 2018.

[14] Vader Github repository. https://github.com/cjhutto/vadersentiment, 2018.

[15] Python Library (KFold). http://scikit-learn.org/stable/modules/classes.html#module-sklearn, 2018.

[16] Herbert Jaeger. Machine learning lecture notes, May 2018.

[17] Numpy Python Library. http://www.numpy.org/, 2018.