Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ETH Zürich

Department of Management, Technology, and Economics
Chair of Entrepreneurial Risks

Master Thesis

The Interplay between Social Media Sentiment and the Cryptocurrency Market

Author:

Lukas Eisner

Supervisors:

Prof. Dr. Didier Sornette
Dr. Ke Wu

May 13, 2019

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

The interplay between Social Media Sentiment and the Cryptocurrency Market

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Eisner	Lukas

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Lucerne, 13.5.19	L. Eisner

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Abstract

We conduct an extensive Social Sentiment Analysis by gathering and analyzing Twitter data for 20 of the larger cryptocurrencies by market volume in two forms: A Multi-Class Analysis classifies individual tweets into the three categories: *positive*, *neutral* and *negative* while an Interval Classification distinguishes a more distinct measure on a scale between *extremely positive* to *extremely negative*.
For the Multi-Class Sentiment Analysis, a Random Forest model is chosen after manually labelling a dataset of 1500 tweets and training as well as evaluating multiple supervised learning models. The Random Forest model achieves a classification accuracy of almost 90% on the test set of our manually labelled data.
The Interval Classification leverages an existing lexical resource (SentiWordNet) and its thresholded values reaches an accuracy of 60% on our labelled data.

Both Sentiment Analysis approaches are further processed and then compared to the financial performance of the respective cryptocurrencies. A lead-lag analysis is performed by means of lagged cross-correlation evaluation and the Thermal Optimal Path (TOP) method. It is concluded that the lead-lag dynamic between social sentiment and financial performance is time-varying and not equal for all cryptocurrencies. While more main-stream coins like Bitcoin (BTC) and Etherium (ETH) show similarities, smaller altcoins like Ripple (XRP) seem to have a more constant lead-lag dynamic where social media sentiment is leading and market performance is lagging behind.

Contents

1 Introduction **1**

2 Data Acquisition and Preparation **2**
2.1 Twitter Scraping . . . 2
2.1.1 Implementation of Twitter Scraping . . . 2
2.2 Filtering Tweets . . . 4
2.2.1 Avoiding Duplicates . . . 5
2.2.2 Finding Inclusion Keywords with the GloVe Method . . . 5
2.2.3 Keywords to exclude ticker-tweets and non-ASCII symbols . . . 6
2.3 Sentiment Analysis specific Preprocessing . . . 8
2.4 Financial Data . . . 9
2.4.1 Financial Data Acquisition . . . 9
2.4.2 Financial Data Visualisation . . . 9

3 Sentiment Analysis **12**
3.1 Multiclass Classification from Scratch (Method 1) . . . 12
3.1.1 Manual Labelling of Data Sample . . . 12
3.1.2 Models . . . 13
3.1.3 Evaluation . . . 16
3.2 Interval Labelling based on SentiWordNet (Method 2) . . . 21
3.2.1 Evaluation . . . 21
3.3 Comparing Method 1 (Random Forest) and 2 (SentiWordNet) . . . 23
3.4 Sentiment Data Post-Processing . . . 24
3.4.1 Interval Aggregation . . . 24
3.4.2 Missing Data . . . 25
3.4.3 Weighted Sentiment . . . 26

4 Statistical Analysis **27**
4.1 Data Preparation . . . 27
4.1.1 Stationarity . . . 27
4.1.2 Rolling Mean . . . 28
4.2 Lead-Lag Analysis . . . 30
4.2.1 Lagged Cross-Correlation . . . 30
4.2.2 Thermal Optimal Path (TOP) Method . . . 31

5 Conclusion and Outlook **35**

Bibliography **36**

1 Introduction

It has been a long-term fascination of humans to try and predict what the future will hold (Hawkins 2004). After going through multiple poorly working methods in the earlier stages of humanity like analyzing flight paths of birds and reading palms, better and more scientific methods started to be developed. Those more advanced methods all have one thing in common though: They need a lot of data to work. This is where social media comes in. It provides researchers with tons of freely available data on a diverse range of topics and has been successfully utilized to predict future developments in a variety of areas like epidemiology (Lampos & Cristianini 2010), politics (Gayo-Avello 2013) and most interesting for this thesis, stock market movement prediction (Bollen et al. 2010) as well as (Guo et al. 2017).

Cryptocurrency has had an increasing relevance for financial transactions since the inception of Bitcoin in 2009. While originally used mainly for illegal transactions due to *"Purported anonymity, ease of cross border transport, lack of clear regulations and settlement finality"* (Vigna & Casey 2016), Bitcoin has since risen to become a digital commodity of interest whose worth is arguably comparable to traditional fiat currency (Colianni et al. 2015).
But Bitcoin still has its problems: High transaction fees, non-instantaneous transactions, waste of energy due to mining (generating Bitcoin) and high volatility (to only name a few). Due to this, a plethora of "altcoins" (A combination of the words 'alternative' and 'coin') has been developed: They are all intended to tackle one or more of these problems. Some better known Altcoins are Etherium (ETH), Ripple (XRP) and Litecoin (LTC). Together with thousands of other cryptocurrencies, they build a market which is not totally dissimilar to the traditional stock market.
The intent of this thesis is to see if social media sentiment could be used to make predictions for the cryptocurrency market similar to what Bollen, Mao and Zeng (Bollen et al. 2010) did for the traditional stock market.

2 Data Acquisition and Preparation

Sentiment information can be gathered from a multitude of sources. When focusing on the digital public space, it is commonly found in both RSS feeds and by analyzing social media. This thesis focuses on the second part, and more exact, on Twitter sentiment. Working with such large quantities of available data makes it strictly necessary to automate the process of gathering individual relevant tweets. This can be done by either relying on the public Twitter API (Application Programming Interface) or by scraping (See chapter 2.1). Since the public Twitter API only supports a historical tweet search of up to 7 days into the past (Twitter-FAQ 2019) and the planned analysis requires a much larger time-frame, scraping is the only valid option.

2.1 Twitter Scraping

Web Scraping is *"a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format"* (WebHarvy 2019). It can be necessary for data gathering when the provided API of a website is not good enough for the task at hand. The most well known case of Web Scraping is the scraping which common search engines like www.google.com perform by using crawlers to index web content. Because being found by search engine queries is extremely relevant for almost all websites, most of them have decided to generally allow users to scrape their website, as long as they behave appropriately and don't slam the server with too much traffic and requests. In case the scraping performed is too harsh, websites often blacklist the origin IP (Internet Protocol) address of the scraper for a certain time-frame. In the case of Twitter, this blacklist duration seems to be around 24h. IP address blacklisting can be circumvented by using proxy servers or even better completely avoided by complying to the website's crawling policies.

2.1.1 Implementation of Twitter Scraping

A very useful project hosted on GitHub called GetOldTweets-python (Jefferson-Henrique 2018) is used as a base for scraping twitter while bypassing the aforementioned time constraints of the official Twitter API. Since this python project mainly supports single search term criteria and has no convenient output format, we have to adjust and automate it to reliably scrape twitter for multiple specific keywords and time-frames. To avoid blacklisting, web crawling best practices have to be followed which includes auto throttling mechanisms while scraping. To find relevant tweets about cryptocurrency as a whole as well as specific coins, a list of the top 19 Altcoin by market cap in September 2018 and their respective abbreviation is compiled. Combined with "#BTC" for Bitcoin, this provides a total of 20 search terms (See Table 1). Two other search term options were considered, but using the full coin name as search query results in significantly less scraped tweets and using the abbreviation without the # sign provides a lot of unrelated tweets which just happen to have the search term as part of a word or sentence.

Table 1: Table of the top 20 cryptocurrency coins by market cap and their *Hashtag* abbreviation in September 2019.

Coin	Twitter abbreviation
Cardano	#ADA
Bitcoin Cash	#BCH
Bytecoin	#BCN
Bitcoin	#BTC
Dash	#DASH
EOS	#EOS
Etherium Classic	#ETC
Etherium	#ETH
ICON	#ICX
IOTA	#IOT
Litecoin	#LTC
NEO	#NEO
Qtum	#QTUM
TRON	#TRX
VeChain	#VEN
NEM	#XEM
Stellar	#XLM
Monero	#XMR
Ripple	#XRP
Zcash	#ZEC

To gather as much data as possible, twitter is scraped for these search terms for the time-frame starting on 01.01.2013 until 27.07.2018. Of course, many of the smaller Altcoins have only been created in the last two years, so it is expected that there would be only very few tweet scrapes for many coins in the early years. This is also a good test to check for "false positives", which means that some of the *Hashtags* might not be solely used to describe a cryptocurrency. This is further discussed in section 2.2: Data Preprocessing.
The final result is a collection of around 33'800'000 tweets (or 6.5gb of raw .csv data) in the following format:

Table 2: Example on the format of a scraped tweet.

Coin	Text	ID	Exact Date	Retweets
#BTC	I love #BTC	1234567	01/01/2018 01:01:01	10

Naturally, this raw data contains a lot of useless or even misleading information. Therefore, a rigorous preprocessing is necessary before applying a sentiment analysis (See Section 2.2: Filtering Tweets).

2.2 Filtering Tweets

It is important to have a clean dataset when approaching Sentiment Analysis. Therefore, the data has to be preprocessed properly. This includes both general preprocessing (like filtering out duplicates, irrelevant tweets and tweets with non-ASCII symbols) as well as Sentiment Analysis specific preprocessing (converting the text into numeric values which a computer can process better). Table 3 shows the amount of tweets for every coin for each filtering step as well as the percentage of tweets remaining after the whole filtering process. The individual filtering steps are described in the next sections.

Table 3: Table which shows the amount of tweets for each coin for every step of the filtering process as well as the percentage of tweets remaining after filtering.

Coin	Unfiltered Tweets	Removed Duplicates	Including Word Filter	Excluding Word Filter	Relevant Tweets
Cardano	560409	504480	241726	112530	20.08%
Bitcoin Cash	326542	299986	254248	129627	39.70%
Bytecoin	1168484	926525	141075	45472	3.89%
Bitcoin	11389796	5087284	5086683	2487657	21.84%
Dash	1342236	1003986	263615	138694	10.33%
EOS	573585	528325	292551	148901	25.96%
Etherium Classic	633017	557524	193065	92109	14.55%
Etherium	2151046	1658537	1499941	720025	33.47%
ICON	42776	40771	29802	14063	32.88%
IOTA	10555810	9513361	3254599	1839178	17.42%
Litecoin	1121370	947365	844684	452868	40.39%
NEO	541284	464593	273812	134255	24.80%
Qtum	44897	40946	34499	18971	42.25%
TRON	862720	702463	400155	154748	17.94%
VeChain	500078	420108	39415	17154	3.43%
NEM	232741	197784	147664	43658	18.76%
Stellar	211704	196000	175615	73345	34.65%
Monero	174523	156809	142900	87873	50.35%
Ripple	1291707	1081832	856002	372032	28.80%
Zcash	74667	65660	49438	33196	44.46%

As an example, Table 4 shows five tweets and the decision for each tweet if they are relevant for the further analysis.

Table 4: An example set of tweets and the decision and reasoning on whether they are relevant or should be filtered out.

Coin	Tweet	Decision
BTC	Duplicate skilled traders automatically with Bitcoin! I copy "wangzai888" Do you? Talk about a nice & smooth 'set-and-forget' commission! # BTC	Keep
BTC	3'081 S.DICE @0.00370000BTC = 11.39970000BTC (= 151.73 USD). 64 # trade # bitcoin # btc # stocks	Remove because it contains an Exclusion Keyword
BTC	Ø´Ù‡Ø± Ù^Ø§ØØ¯ ÙÙ‚Ø · .. ÙŠÙØµÙ„Ù†Ù†ÙŠ Ø¹Ù† Ù…Ù^Ø¹Ø¯ Ø³Ø³Ù„Ù„Ù… Ø¨Ù…Ø§Ø´Ø±Ø© Ù„Ù„ÙŠÙˆÙ… .. Ø£ØµÙ„Ù§ Ø¹Ø§Ø¯ÙŠ :) # btc	Remove because non-ASCII icons (Most likely non-latin language)
IOT	# iot IoT - Internet of Things	Remove because it doesn't contain any Inclusion Keywords
BTC	Duplicate skilled traders automatically with Bitcoin! I copy "wangzai888". Do you? Talk about a nice & smooth 'set-and-forget' commission! # BTC	Remove because it is a duplicate

2.2.1 Avoiding Duplicates

At first sight, it might seem sufficient to just delete all tweets (but one) which have the same *ID*. But since tweets can be retweeted (and this is already reflected in the *Retweets* column), it is also necessary to filter for identical *Text*. Otherwise, all retweets would be counted twice.

2.2.2 Finding Inclusion Keywords with the GloVe Method

Next, it is important to only consider tweets which are actually discussing the cryptocurrency space. A good example of a "false positive" tweet would be a tweet talking about the Internet of Things IoT. It gets caught by the scraper because the IOTA coin has been abbreviated with #IOT before switching to the more unique #MIOTA. Other examples include #BCN (also used to abbreviate the city of Barcelona) and #VEN (used to describe the country of Venezuela). To filter tweets out which are not talking about cryptocurrency, a list of words relevant to the cryptocurrency space is necessary.

To gather the most relevant keywords while avoiding pure cherry-picking, a pre-trained *Global Vectors for Word Representation* (GloVe) model (Pennington et al. 2014) is used. Its Euclidean distance between two word vectors measures their linguistic and semantic similarity. The model is applied to the three words *cryptocurrency*, *bitcoin* and *currency*. The top 30 results (the 30 shortest Euclidean distances) after excluding duplicates can be seen in Table 5. To broaden the scope, the same method is now applied to those 30 top results (this time only checking for the 10 most relevant similar words to each of them). After removing duplicates, this results in a list of 216 inclusion keywords. It is important to note that one of the 216 keywords generated is

BTC itself which explains the low number of filtered out tweets in the *Inclusion Word Filter* process for *#BTC*.

Table 5: Table of the 3 cherrypicked words and the 30 most relevant similar words for the cryptocurrency space according to a pre-trained GloVe model.

Inclusion Keywords
currency
cryptocurrency
bitcoin
encryption
keynesian
strategists
currencies
decentralized
algorithm
crowdsourcing
foundations
crowdsourced
commodity
speculate
exchange
trading
investment
markets
monetary
investing
debt
banking
wealth
speculating
futures
stock
shares
forex
stocks
traders
trades
bullish
bearish

2.2.3 Keywords to exclude ticker-tweets and non-ASCII symbols

Ticker-tweets are automated tweets which post the current cryptocurrency price at a high interval. Their influence on the actual sentiment of people is negligible at best and could even be misleading because of their high frequency when using automated sentiment analysis. To

filter out such tweets, the data is scanned manually for obvious ticker tweets. Their recurring pattern (such as *"sdice"* which stands for a cryptocurrency gambling website which offers an hourly ticker) is used to filter the data (See Table 6). We also add a regular expression at the end of the table to filter out non-ASCII symbols.

Table 6: List of 10 typical patterns for ticker tweets as well as the regular expression for non-ASCII symbols.

Exclusion Keywords
sdice
smpoe
S.Dice
S.MPOE
=
bitcoinde
faucet
bitcoinprice
arb opps
spinner
$[\hat{}\backslash x00 - \backslash x7F]+$

2.3 Sentiment Analysis specific Preprocessing

Natural Language Processing (NLP) is a scientific field which is concerned with the translation of natural human language into something a computer can easily work with. Its origins go all the way back to the 1960's (Schank & Tesler 1969). Until the late 1980s, this translation was mainly rule-based. But with the statistical revolution (Johnson 2009) in the early 1990s, the approach changed fundamentally to rely more heavily on machine learning.
In this thesis, NLP approaches are used to translate tweets into multiple Tokens (not to be confused with crypto-tokens) which can then in turn be individually analyzed and rated based on their sentiment. Specifically, a Tokenizer implementation from Keras (Chollet et al. 2015), which is an open-source neural-network library for python, is used to split phrases or sentences into individual tokens. Then, a bag-of-words model simplifies these tokens by calculating the frequency of occurrence of each token in the whole dataset and finally returning a vector for each tweet which contains said frequencies. Since not all tweets have the same length, the vector length is also padded to the maximum amount of tokens a tweet can have. This format is a lot more suited for further computer-based analysis. An example tweet can be found in Table 7.

Table 7: A simple example on how a tokenizer and a bag-of-words model are used to represent a tweet in numerical form.

Tweet	bitcoin price looks north as stock market falls again # btc									
Tokenized Tweet	bitcoin	price	looks	north	as	stock	market	falls	again	btc
Occurance Rating	1	28	498	581	53	317	40	1426	236	2
Returned Array	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,28,498,581,53,317,40,1426,236,2]									

2.4 Financial Data

2.4.1 Financial Data Acquisition

The financial information about all cryptocurrencies is gathered with the help of the CryptoCompare.com API (Cryptocompare 2018). It allows easy access to hourly prices of all relevant cryptocoins since their creation.

2.4.2 Financial Data Visualisation

Since it would be too much to plot the financial data for all top 20 altcoins, we will only have a look at Bitcoin, Etherium and Ripple. They represent three of the larger cryptocurrency coins by market value. Their individual coin value differs by several magnitudes. We will also narrow our focus on a 1-year window from August 2017 until August 2018, which includes the big rise and fall of cryptocurrency in Winter 2017/2018: Figure 1, 2 and 3 all clearly show the cryptocurrency hype in the end of 2017 and the brutal crash soon after.

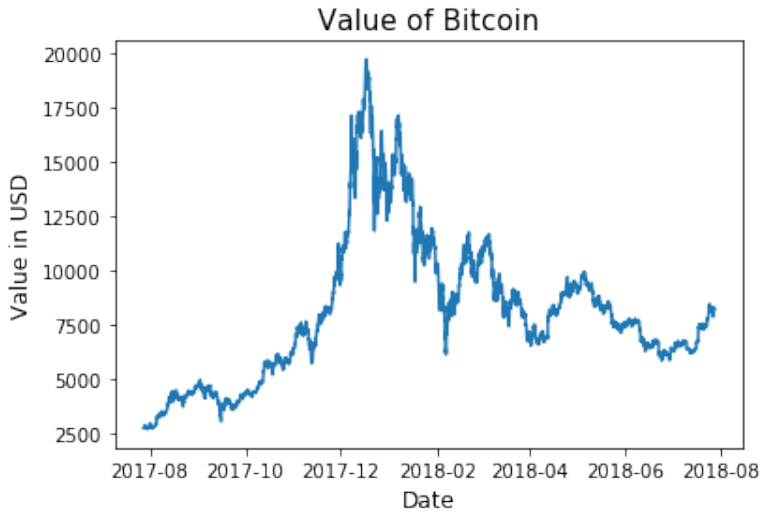

Figure 1: The Value of Bitcoin in USD for a 1-year window.

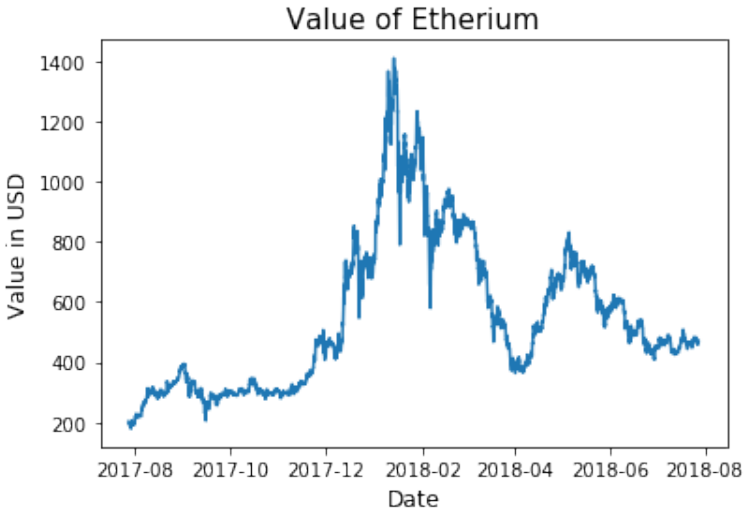

Figure 2: The Value of Etherium in USD for a 1-year window.

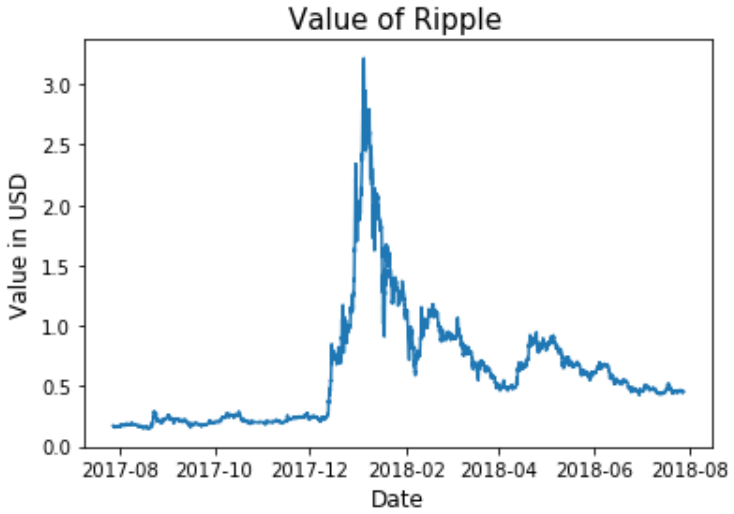

Figure 3: The Value of Ripple in USD for a 1-year window.

It is often advisable to look at the daily (or hourly) change in price instead of the absolute value. Therefore, we also show the daily change (delta) in Figure 4, 5 and 6. We can easily see the larger daily fluctuations around the hype period.

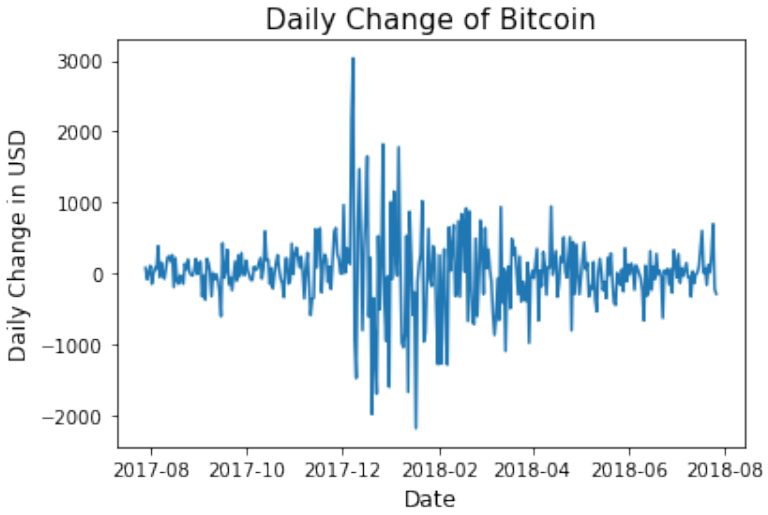

Figure 4: The daily change of Bitcoin in USD for a 1-year window.

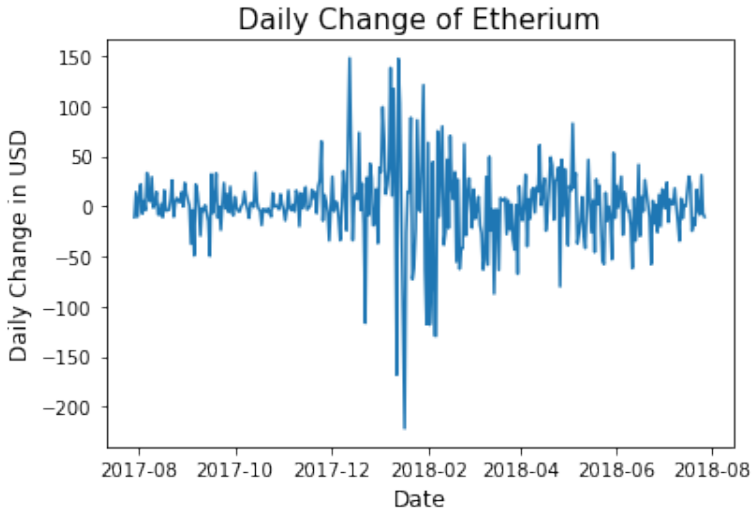

Figure 5: The daily change of Etherium in USD for a 1-year window.

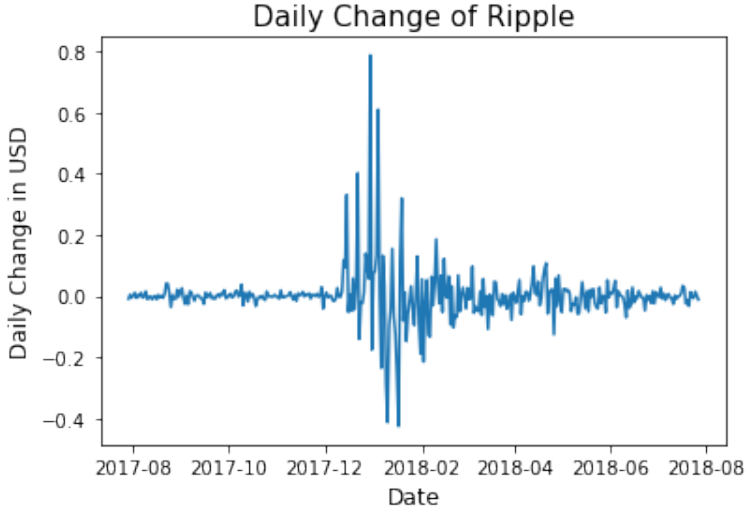

Figure 6: The daily change of Ripple in USD for a 1-year window.

3 Sentiment Analysis

Sentiment Analysis is a broad term and can mean anything from a pure manual examination of a written (or spoken) text to a fully automated, multidimensional analysis of data from a plethora of sources. Because of the magnitude of our data, we will focus on an automated approach in this thesis. A common use of Sentiment Analysis is to classify a text (or chunk of text) as *positive* or *negative*. But other classifications are possible to: Text can for example also be categorized as *objective* vs *subjective* or *backward looking* vs *forward looking*. While irony and sarcasm were a large problem for sentiment classification, newer approaches often succeed in handling them well (Farias & Rosso 2017).
We will explore two methods of Sentiment Analysis: Method 1 will try to classify each tweet as *positive*, *neutral* or *negative* by building and training an appropriate machine-learning model, while Method 2 will classify the tweets along a spectrum from *totally negative* to *totally positive* by using a pre-trained model based on SentiWordNet (Baccianella et al. 2010).

3.1 Multiclass Classification from Scratch (Method 1)

A machine learning model works differently than a conventional statistical model: Instead of requiring an input and a specific mathematical formula to get the output we are looking for, we instead provide the machine learning model with both input and output and let it decide on the optimal calculations necessary to get there.
For this reason, we first need a so called *labelled* data set. This consists of both the original tweet and a manual label for the sentiment. In our case, this label is either "-1" for *negative*, "0" for *neutral* or "1" for *positive*.

3.1.1 Manual Labelling of Data Sample

To optimally train a machine learning algorithm, a balanced training set is preferable. This means that each of the different classes appears with the same frequency in the manually labelled tweets. Therefore, a random sample of tweets is drawn and labelled until there are 300 negative, neutral and positive labelled tweets respectively. To improve the accuracy of the models, 200 additional (highly) negative, neutral and positive tweets are later added which have been produced by a first iteration of our Random Forest model (See Chapter 3.1.2) and have been manually verified (We will refer to these tweets as *semi-manually labelled* from now on). This provides us with a total labelled dataset of size 1500.
All tweets are labelled with the following question in mind: *Would this tweet make me see the specific cryptocurrency coin in a positive / neutral / negative light?*. For this reason, tweets which simply mention that a coin has a specific value without mentioning a trend are considered *neutral*, while for example a tweet specifically stating a downward trend of a coin or a technical problem like hacks or breaches will be considered *negative*.
A sample of tweets labelled in this manner can be seen in Table 8.

Table 8: An exerpt from the list of 1500 manually labelled tweets where "-1": *negative*, "0": *neutral* and "1": *positive*

Tweet	Label
How happy are you with your # PrivateBank ? > http://bit.ly/2s4aguc # sks8 # markets # bank # btc # rt # mkt pic.twitter.com/V4tub8YxxT	**0**
Are you part of BeetleCoin family? # cryptonews # cryptolife # cryptoinvestor #cryptocurrencynews # BTC # Blockchain # masternode # altcoins # staking #passiveincomepic.twitter.com/vi7E2TTcEy	**0**
The Hardware Bitcoin Wallet. Get Trezor now for only $99 https://buytrezor.com?a=coinokbuytrezor.com/ ?a=coinok # btc # bitcoin18pic.twitter.com/Si86SSGEjq	**0**
Simon Dixon: 'Bitcoin Solves 3 Major Problems in the Financial System' http://a.givemeasay.com/PYo # btc # bitcoin	**1**
Swiss town of Chiasso to accept bitcoin to settle tax bills.: http://ift.tt/2eRTVaj #bitcoin # btc	**1**
How crypto-currencies like bitcoin have taken roots in Indias " EconomicTimes http://bit.ly/2zVE00F # bitcoin # fintech # btc # crypto	**1**
Hacked: http://BitcoinTalk.org User Data Goes Up For Sale On Dark Web http://bit.ly/2cyTCvz # BTC # bitcoin	**-1**
Market Crash: Bitcoin [BTC] to fall below 6000? https://ambcrypto.com/market-crash-bitcoin-btc-to-fall-below-6000/ BTC ETH *ETH#Ripple* # BTC # Bitcoin # Ethereum $ xrp # BitcoinCash #Altcoins	**-1**
Bitcoin prices have been manipulated, study says # btc # altcoin # fintech https://coinspectator.com/news/527901/bitcoin-prices-have-been-manipulated-study-says	**-1**

3.1.2 Models

There are many different models which can be applied to a problem such as this. And they can be tuned very differently as well (which means their hyperparameters can be modified to improve the performance of the model). In the following sections, 7 different approaches will be presented. Afterwards, their confusion matrices (See Chapter 3.1.3) will be used to compare the usefulness and accuracy of the different models and a winner will be picked.

Naive Bayes Model

Naive Bayes is a classifier built on the application of the Bayes' theorem (1), which states how we can find the probability of **A** happening, given that **B** has occurred . The classifier is called naive because it assumes that all features (or individually measurable properties or

characteristics of the input) are independent.

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)} \tag{1}$$

The classifier works by building a *Likelihood Table* which reflects how frequent a feature turns up in each label. It then applies the Bayes' theorem to calculate the posterior probability for each class and the class with the highest posterior probability is the prediction of the model.
A big advantage of the Naive Bayes Model is the fact that it is very easy and fast. Therefore, it is used frequently in real-time prediction and can outperform more sophisticated models in both speed and accuracy depending on the task at hand. It is also commonly used as a base-line model, i.e. it provides an initial performance score which can be used to evaluate the benefit of more complex approaches.

Logistic Regression Model

The *Logistical Regression Classifier* is generally used for a binary classification problem and is part of the GLM (*Generalized Linear Model*) family. It uses the Maximum Likelihood Estimation algorithm to predict the probability of the dependent variable (label). The name of the model comes from the fact that a Logit transformation (2) is performed to linearize a sigmoid distribution before calculating the regression equation which in turn is then used to predict the probability of a certain outcome of the dependent variable.

$$Logit = log(\frac{P(A)}{P(\overline{\mathrm{A}})}) \tag{2}$$

Since we are dealing with a multiclass problem instead of a binary problem, we apply the generalized *Multinomial Logistic Regression* with a multinomial distribution assumption.

Classification Tree

A *Classification Tree*, also called *Decision Tree*, is a visual representation of a series of decisions and was developed by Breiman et al. (1984).
We will illustrate this concept on a fictional example: Let there be a list of 100 people which should be binary classified as expected to be *rich* or *poor*. We know for each person if they had a *good education* or *bad education* and if they are at $age < 15$ or $age >= 15$. We know their classification (label) beforehand and want to build a tree with a high classification accuracy. We will start at the root of the tree in Figure 7. To decide which feature to use first as decider, the *Gini Index* (Ceriani & Verme 2012) for every possible feature is calculated and the feature which provides the best separation of the original data into distinct subgroups (according to their label) is chosen.
In our case, the decider *age* is chosen and the list is split into two groups of people depending

on their age. For the group with $n = 20$, our model predicts that the people are poor. We now compare this fact with our initial labelling and decide that it is mostly correct. We therefore won't continue splitting up the $n = 20$ people into further subgroups.
For the node with $n = 80$, we don't have such a high accuracy. We therefore perform the same algorithm again, this time with *education* as decider. We finally reach a spot where our model has a high accuracy when predicting whether the $n = 100$ people are *rich* or *poor*.
Now that the tree is constructed, it can be easily applied to a new set of people by sending them through the decision tree and labelling them accordingly.

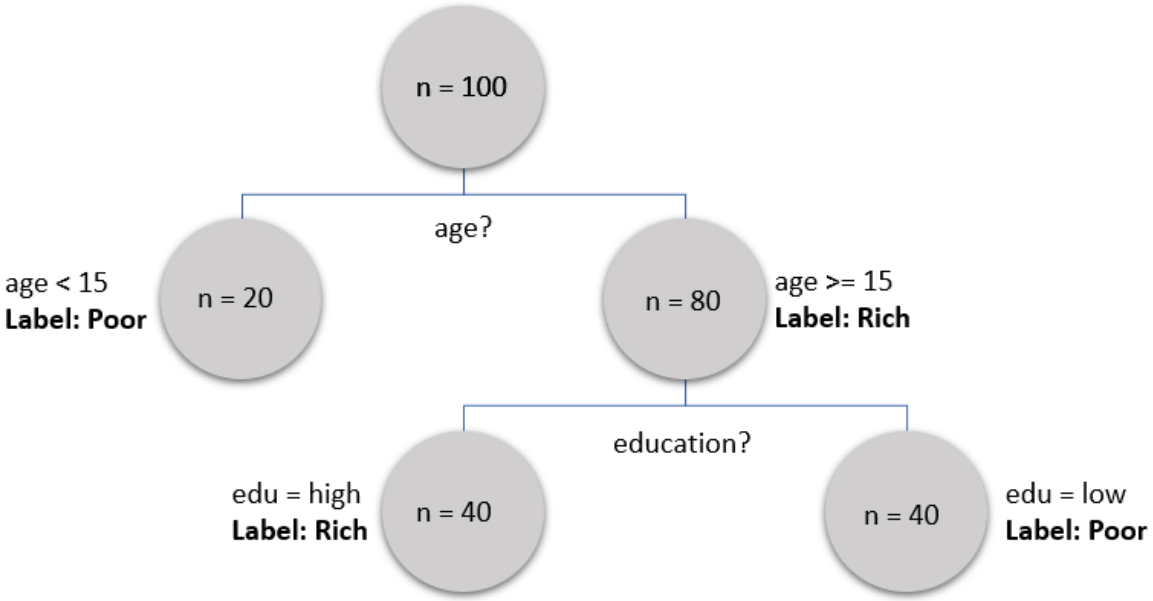

Figure 7: Example of a simple classification tree with n denoting the people in each node, *age* and *edu* denoting features and finally *Label* showing the decision.

Random Forest

Random Forest is an algorithm which builds on *Classification Trees* and was similarly developed by Breiman et al. (1984). While a *Classification Tree* is only performed once, a *Random Forest* is an ensemble classifier based on multiple versions of a *Classification Tree* with slightly different input and feature sets.
To get these modified input and feature sets, a random subset of size $n = \frac{2}{3}N$ (in our case) of the input is taken. On this subset, *Bagging* is performed. It denotes the selection of 'm' features from the total 'M' features (in our case: $m = \sqrt{M}$). A classification tree is built with the input subset of size n and the feature subset of size m. This is repeated k times and each tree is evaluated with the remaining input of size $\frac{1}{3}N$ and finally the best performing *Classification Tree* is selected.
A big benefit of the *Random Forest* approach is the fact that the multiple random steps inhibit the overfitting problem (learning the data by heart) and reduce the variance at the same time: Formula 3 (Hastie et al. 2001) shows the average variance of k trees. While it is obvious that

increasing the number of trees k make the second part of the equation shrink away, we need to reduce the correlation between the trees to reduce the first part of the equation. This effect can be achieved by only picking a random subset of the M features in the data at each node split. Every single tree is now forced to use different predictors for splitting at every node. Thereby, we now do not only construct different trees but also more de-correlated ones.

$$var_{all\ trees} = \rho\sigma^2 + \frac{1-\rho}{k}\sigma^2 \tag{3}$$

Gradient Boosting Model

Gradient Boosting refers to an iterative approach where a tree $n+1$ is fit on the errors of tree n (This is where the word *Gradient* comes from which reflects that error). The initial tree can be produced as a simple classification tree. It was first proposed by Friedman (2001).
Unlike the models we covered before, this is the first model where one can say that it is actually "learning" since the performance is evaluated by a loss function in each iteration which in turn has to be minimized.

Support Vector Machine

A *Support Vector Machine* classifier plots each data point (in our case tweet) in an m-dimensional space where m is the number of features from the dataset. It then performs a binary classification by finding the linear hyper-plane that differentiates the two classes the best.
Since we are dealing with a multi-class problem, a "on-against-one" (Knerr et al. 1990) approach is chosen to pit each class against one another individually. Finally, a majority vote over all "one-on-one" classification is taken to decide on the final label.

Convolutional Neural Network

Explaining a *Convolutional Neural Network* in detail would go beyond the scope of this thesis. Therefore, we will only say that a CNN is a type of neural network which is widely used for grid-like topologies such as images (2d or 3d grids) and time-dependant data (1d grid e.g. time-series). Its feature extraction proficiency can reveal relationships which remain hidden to the classical approaches discussed before. But due to its iterative nature and tendency to overfit, it needs a larger labelled dataset to train it to its maximum performance.

3.1.3 Evaluation

All classification methods are applied to the manually labelled tweet sample. To evaluate their performance, one should have a look at the confusion matrix. To illustrate this concept, the confusion matrix for a simple binary classification model can be found in Figure 8. Tweets with a positive sentiment which the model labelled correctly are denoted with TP (True Positive).

Similarly, tweets with a negative sentiment which are labelled correctly are denoted with TN (True Negative). FP (False Positive) and FN (False Negative) show the cases where the model either labelled a negative tweet as positive or a positive tweet as negative. The confusion matrix provides us with easy-to-calculate measures about the performance of our classification. As an example, (4) shows the calculation for the Accuracy measure.

Confusion Matrix 2x2		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 8: The confusion matrix for a binary classification. **TP** (True Positive) is the number of tweets which are labelled as *positive* and are actually positive (according to the manual labelling). The number of tweets which are **TN** (True Negative), **FP** (False Positive) and **FN** (False Negative) are calculated analogously.

$$2x2\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Our problem is slightly more complex though: Since we have 3 classes, the confusion matrix will be of shape 3x3. Since we added the "Neutral" classification, it is easier and for our case sufficient to simply label each classification as either "True" or "False" as seen in Figure 9. The accuracy once again calculates by adding up all the correctly labelled tweets and dividing this by the total number of tweets.

Confusion Matrix 3x3 (Simplified)		Actual Values		
		Positive	Neutral	Negative
Predicted Values	Positive	True	False	False
	Neutral	False	True	False
	Negative	False	False	True

Figure 9: The simplified confusion matrix for a 3-class classification.

We now apply our models from 3.1.2 to our manually labelled dataset of 1500 tweets. To avoid our models learning the rather small dataset by heart (*overfitting*), we split our data into two parts: The *Training Data* and the *Validation Data*. With a split of 66% to 34%, this leaves 990 tweets to train our models. We then evaluate the model by applying it to the *Validation Data*, which it has never seen before.

Figure 10 to 16 show the confusion matrices and the calculated accuracy score for each model. Since our data is pretty balanced, we expect a random classification to achieve an accuracy of about 33.3%. Therefore, all models with the exception of the Support Vector Machine seem to perform better than a random classification. Nevertheless, there are big differences in the performance: While a **CNN** and a **Support Vector Machine** might outperform the other models with enough training data, they obviously fails to provide acceptable accuracy with a small dataset. On the other side of the spectrum, a **Naive Bayes** classification seems to simply not be appropriate for the task at hand.
From the well performing models, **Random Forest** has the highest accuracy with 88.4%. An additional benefit of this model is its high computational speed. We therefore choose to continue with this model for our sentiment analysis.

Confusion Matrix **Bayes**		**Actual Values**		
		Positive	Neutral	Negative
Predicted Values	Positive	**117**	69	57
	Neutral	39	**71**	64
	Negative	14	30	**49**
Accuracy Score: 46.5%				

Figure 10: The confusion matrix for the Naive Bayes Model with all **True** classifications in bold.

Confusion Matrix **Log Regression**		**Actual Values**		
		Positive	Neutral	Negative
Predicted Values	Positive	**114**	18	32
	Neutral	3	**60**	53
	Negative	53	92	**85**
Accuracy Score: 50.8%				

Figure 11: The confusion matrix for the Logistical Regression Model with all **True** classifications in bold.

Confusion Matrix Classification Tree		Actual Values		
		Positive	Neutral	Negative
Predicted Values	Positive	**130**	5	36
	Neutral	6	**151**	5
	Negative	34	14	**129**
Accuracy Score: 80.4%				

Figure 12: The confusion matrix for the Classification Tree Model with all **True** classifications in bold.

Confusion Matrix Random Forest		Actual Values		
		Positive	Neutral	Negative
Predicted Values	Positive	**169**	7	46
	Neutral	0	**161**	3
	Negative	1	2	**121**
Accuracy Score: 88.4%				

Figure 13: The confusion matrix for the Random Forest Model with all **True** classifications in bold.

Confusion Matrix Gradient Boosting		Actual Values		
		Positive	Neutral	Negative
Predicted Values	Positive	**168**	17	56
	Neutral	0	**152**	0
	Negative	2	1	**114**
Accuracy Score: 85.1%				

Figure 14: The confusion matrix for the Gradient Boosting Model with all **True** classifications in bold.

Confusion Matrix Sup Vec Machine		Actual Values		
		Positive	Neutral	Negative
Predicted Values	Positive	**5**	0	0
	Neutral	0	**2**	0
	Negative	165	168	**170**
Accuracy Score: 34.7%				

Figure 15: The confusion matrix for the Support Vector Machine Model with all **True** classifications in bold.

Confusion Matrix CNN		Actual Values		
		Positive	Neutral	Negative
Predicted Values	Positive	**126**	46	40
	Neutral	14	**81**	31
	Negative	30	43	**99**
Accuracy Score: 60.0%				

Figure 16: The confusion matrix for the Convolutional Neural Network Model with all **True** classifications in bold.

3.2 Interval Labelling based on SentiWordNet (Method 2)

A WordNet is a lexical database for the English language (Fellbaum 1998) which groups English words into sets of synonyms called synsets. SentiWordNet extends this resource by adding a sentiment score (*positivity*, *negativity* and *objectivity*) to each synset. It is superior to a classical sentiment-labelled dictionary which is based on individual words instead of synsets because the classical dictionary can't differentiate between multiple meanings of the same word. Baccianella et al. (2010) offer an example in their description of SentiWordNet 3.0, the newest iteration of the Lexical Resource: The word *estimable* is part of the synset [estimable(J,3)] in the sense of *"may be computed or estimated"* and can be labelled differently from the same word in the synset [estimable(J,1)] which corresponds to the sense *"deserving of respect or high regard"*.
We use this improved dictionary-based technique to label each individual synset in our tweets with their *positivity*, *negativity* and *objectivity* score. Since *objectivity* can be used interchangeably with *neutrality* in our case (Similar to how we label objective and descriptive tweets as "0" in our manual labelling process), we can again use our labelling scheme of 1 for positive, 0 for neutral and -1 for negative. We then take the mean of all synsets in a tweet to get an interval classification in the interval $[-1, 1]$.

3.2.1 Evaluation

We apply Model 2 to our pre-labelled data for further analysis. Since this pre-labelled dataset is ordered in a manner where the first 500 tweets are negative, the second 500 tweets are neutral and the last 500 tweets are positive, there should be a visible difference of the sentiment score in those 3 intervals. To better visualize this, we also add the mean sentiment score of each interval and split the plot into 3 sections. Figure 17 shows exactly this. But it also shows us that the whole model seems to be biased to rate tweets slightly more positive than it should and that there is a severe difference between the initial 300 tweets and the 200 additional tweets for each label which have been semi-manually labelled (see Chapter 3.1.1).

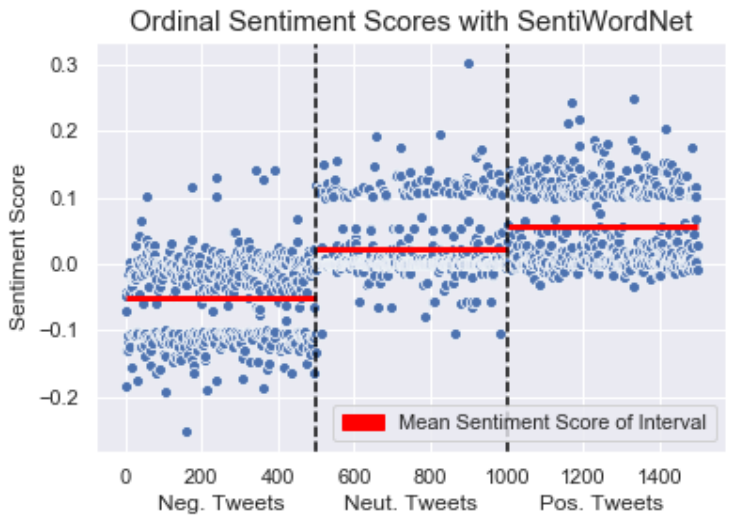

Figure 17: The SentiWordNet scoring system applied to the ordered labelled dataset. The 500 negative tweets have an average SentiWordNet score of -0.05, the 500 neutral tweets have an average SentiWordNet score of 0.02 and the 500 positive tweets have an average SentiWordNet score of 0.06. The horizontal gaps in each section are produced due to the difference between the 300 manually labelled and the 200 semi-manually labelled tweets.

This method is obviously more precise than Method 1 in that it provides us with a float value between -1 and 1 which gives each tweet a sentiment score between *totally negative* and *totally positive*. As an example: A sentiment score of 1 can only be reached if every single synset in the tweet has a positivity score of 1 which is basically impossible since no sentence is constructed with only opinionated synsets. It is therefore more likely to find an overall very positive tweet to have a sentiment score of ~ 0.15.

3.3 Comparing Method 1 (Random Forest) and 2 (SentiWordNet)

Since we can't use a pre-labelled 3-class model to benchmark an interval classification straight up (and therefore be able to compare it to Method 1), we first have to modify our interval classification. We will threshold the interval data to map it to a 3-class model. This means that we will label all tweets with a SentiWordNet sentiment value higher than the threshold as "1" and the tweets with sentiment value lower than the negative threshold as "-1". The tweets in-between the two thresholds are labelled as "0". By choosing a threshold of "0" (and omitting the *neutral* labelled tweets), we can also modify this problem into a binary classification.
We also apply a shift along the y-axis to account for the slight positive bias we have. After iterating over multiple values for both *threshold* and *shift*, we achieve the highest accuracy while maintaining a more or less balanced confusion matrix (which mean a similar amount of *False Positive*, *False Neutral* and *False Negative* classifications) with the value 0.01 for both *threshold* and *shift* (See Figure 18). While the accuracy isn't as good as the one from our RandomForest model, we shouldn't forget that we have additional information due to the interval classification. We will therefore continue with both *Random Forest* from Method 1 and *SentiWordNet* from Method 2 in Chapter 4.

Confusion Matrix SentiWordNet		Actual Values		
		Positive	Neutral	Negative
Predicted Values	Positive	**277**	131	32
	Neutral	161	**308**	102
	Negative	62	61	**366**
Accuracy Score: 63.4%				

Figure 18: The confusion matrix for the SentiWordNet Model with all **True** classifications in bold. Note that the Model has been applied to the whole labelled data set of 1500 tweets instead of only the validation set of length 500.

3.4 Sentiment Data Post-Processing

Now that we have a sentiment value for each tweet (Integer for Method 1 and Float for Method 2), we need to build a time-series from it in order to be able to compare it with our financial data by means of lead-lag analysis.

3.4.1 Interval Aggregation

Since there is little use in only having individual tweets labelled, we aggregate our sentiment data into 1-hour, 6-hour and 12-hour windows (by tweet posting time) for further analysis. As an example, Figure 19 and 20 show the 6-hour aggregation (by means of average) of both Method 1 and 2 provided by our sentiment analysis. Even with such an aggregation, there are still some outliers, but especially with Method 1, a clear pattern of peaks and valleys can be seen and we can also see some movements which seem to interrelate with the Bitcoin value. Please note that the problem of missing data has already been handled according to the method from the next subsection (3.4.2) and that the scale as well as the shift of the y-axis are not indicative of one method outperforming the other but rather expected due to the different sentiment analyses approaches.

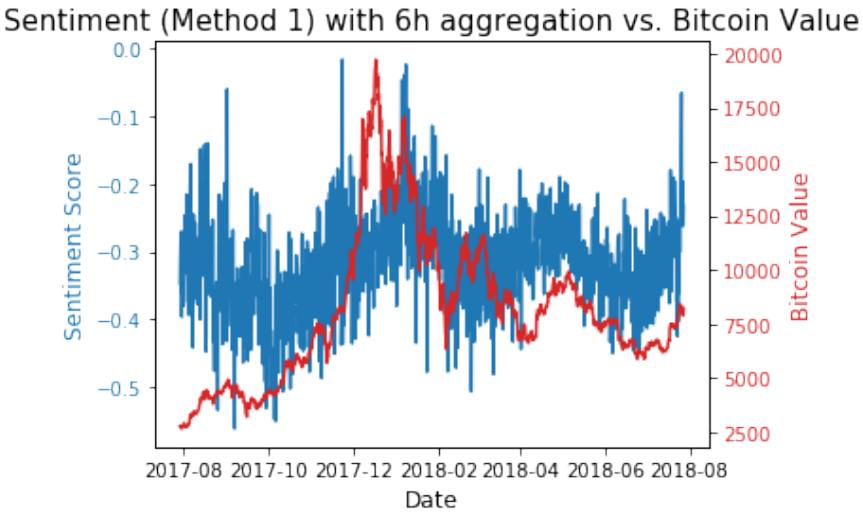

Figure 19: y-axis 1: Sentiment score with a 6h aggregation for the Random Forest approach (Method 1). y-axis 2: Bitcoin value.

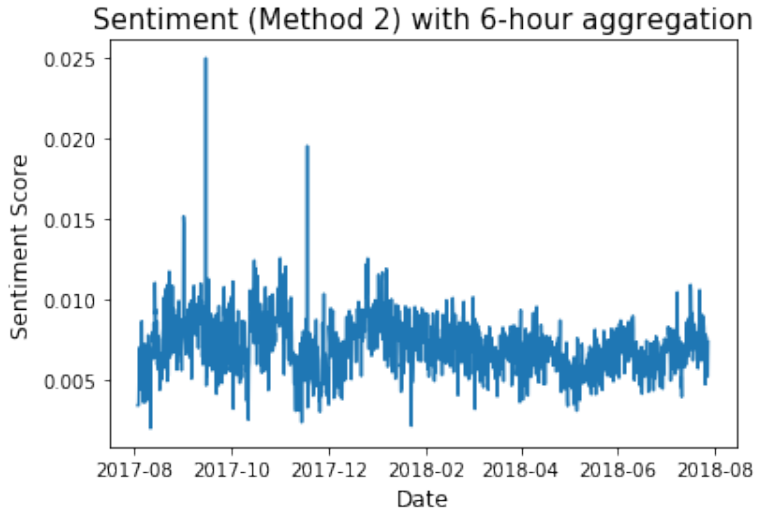

Figure 20: Sentiment score with a 6h aggregation for the SentiWordNet labelling approach (Method 2).

3.4.2 Missing Data

Web Scraping isn't perfect and can sometimes provide incomplete data. In our case, there are several 6-12h windows during our 1-year interval where even multiple iterations of the scraper can't successfully download tweets. This might be connected to Twitter downtimes, but it happens too frequent to only attribute the missing data to this. It might even be a form of scraping protection implemented by Twitter. Whatever the reason, this problem has to be dealt with before continuing further. Figure 21 shows these missing tweets by means of multiple jumps down to the bottom of the y-axis. Since there are no tweets to evaluate, there will also be no valid sentiment score in these periods.

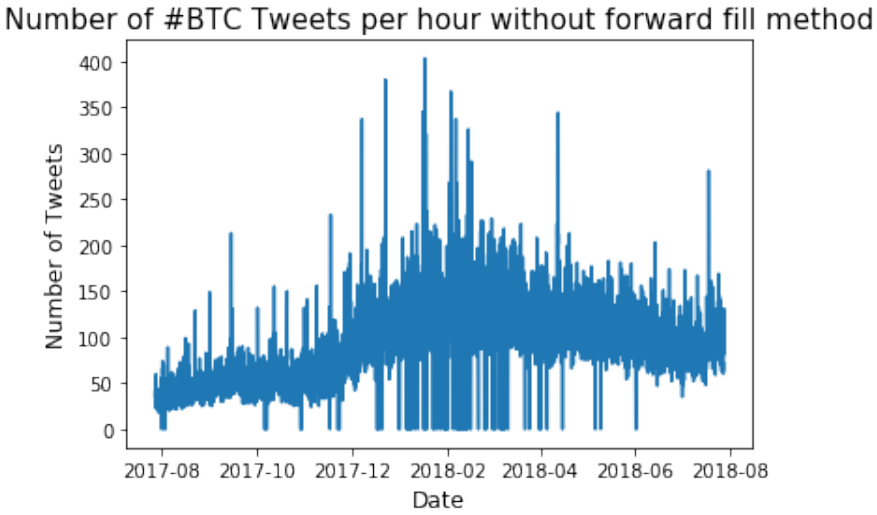

Figure 21: The number of #BTC tweets per hour without the Forward Fill method.

To fix this problem, we employ the Forward Fill (ffill) function on our data. This function looks for missing data and propagates the last valid observation forward. The hereby estimated number of actual tweets can be seen in Figure 22.

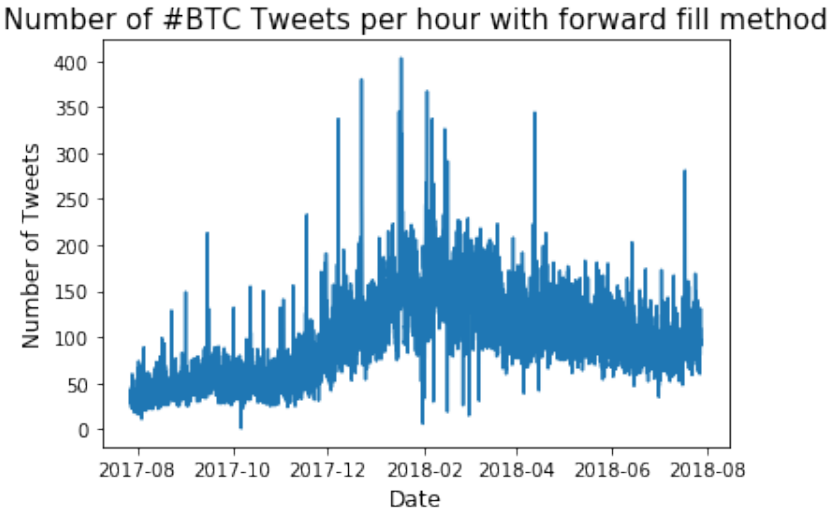

Figure 22: The number of #BTC tweets per hour with the Forward Fill method.

The Forward Fill function has already been applied to our *sentiment* data from Method 1 and 2 in Section 3.4.1.

3.4.3 Weighted Sentiment

We will now have a look at possible benefits and drawbacks on weighing each tweet with its number of retweets. Intuitively, it might make sense to only count each tweet once. But the action of re-tweeting is comparable to comprising your own tweet and therefore inherently carries an opinion as well. Figure 23 shows a comparison of the unweighted vs. weighted sentiment score of Method 1. Notice that weighing our sentiment score based on the number of retweets provides a more "opinionated" sentiment score.
We will therefore continue with our weighted sentiment data (Method 1 and 2).

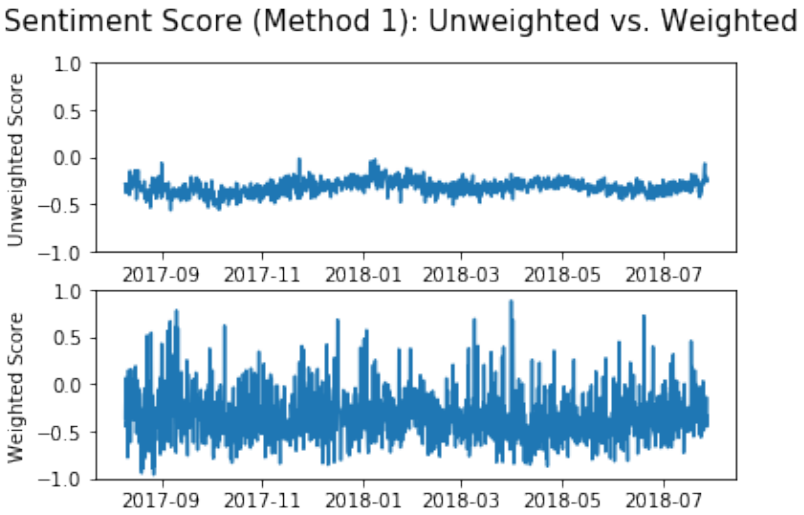

Figure 23: Comparison of unweighted vs. weighted sentiment score for the Random Forest approach.

4 Statistical Analysis

We will now have a look at the interplay between our sentiment data and our financial data. To do this, we first have to prepare our financial data to be in the same shape as the sentiment data. We also have to make sure that all prerequisites for a clean and valid analysis are achieved. We will once again focus on Bitcoin first and then broaden our analysis to other cryptocurrency coins while checking for similarities and differences.

4.1 Data Preparation

Before we can go into correlation and lead-lag analysis, we have to perform some more data preparation.

4.1.1 Stationarity

One prerequisite for most time series analysis is that the time series has to be stationary (which means that statistical measures such as mean, variance and autocorrelation should be constant over time). This can be achieved for example by taking the log of the data or working with the change (delta) between two data points instead of the absolute value (As already shown for the daily change of Bitcoin, Etherium and Ripple in Section 2.4.2). Figure 24 and 25 show these data transformations for the example of our 6-hour aggregated Bitcoin data.

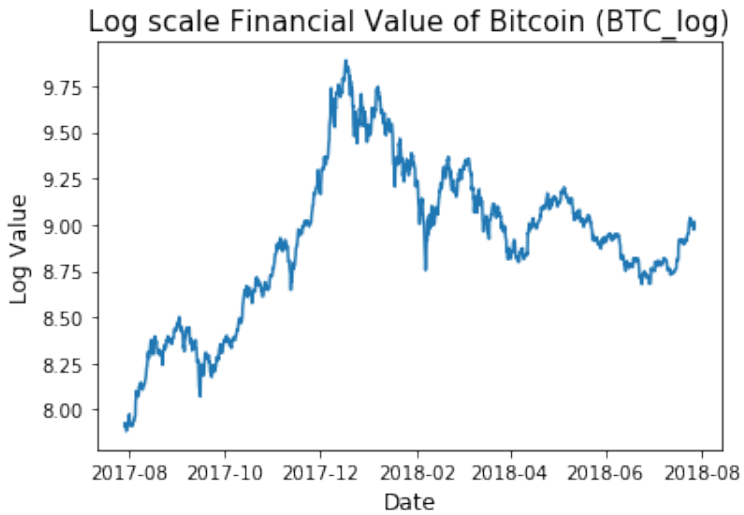

Figure 24: Value of Bitcoin on a logarithmic scale for a 1-year window.

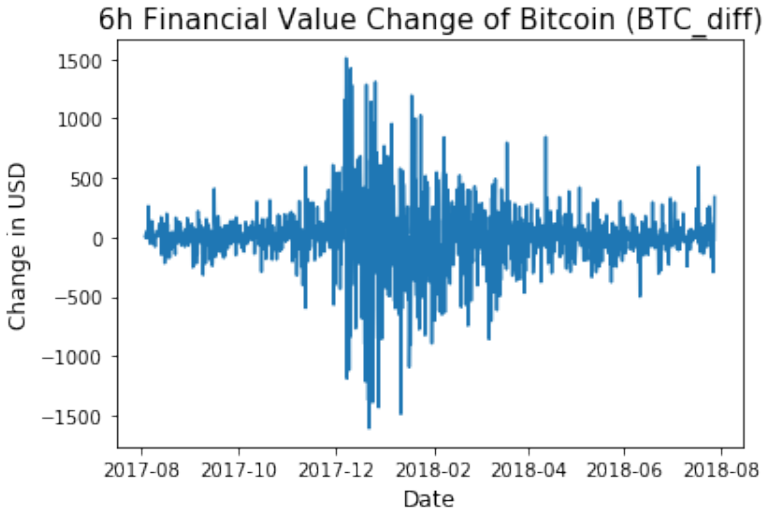

Figure 25: The 6h aggregated change in Value of Bitcoin in a 1-year window.

We now test for stationarity by performing the *Dicky-Fuller* test (Dickey & Fuller 1979) for both approaches. Its null hypothesis is that a unit root is present in the autoregressive model. Therefore, the alternative hypothesis is the stationarity of the time-series.
The test shows non-stationarity for the logged time-series but stationarity for the differenced series. We will therefore continue working with the differenced series *BTC_diff*.
We also apply the same process of calculating the interval change to our sentiment data.

4.1.2 Rolling Mean

To smoothen our sentiment data, we apply a *Rolling Mean* (also called *Moving Average*) to it. This means that we calculate the average sentiment score for a rolling window. This eliminates statistical outliers which might hinder our analysis. It can be seen in Figure 26 for our weighted sentiment score from Method 1 (Random Forest) and in Figure 27 for our weighted sentiment score from Method 2 (SentiWordNet). We choose a *Rolling Mean* window size of 4 which means that each data point overall carries information from a 24h window.

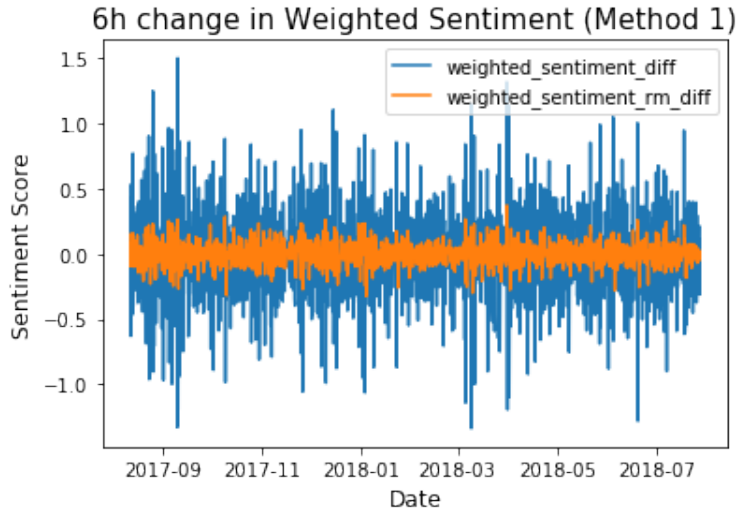

Figure 26: 6h aggregated change in Weighted Sentiment (Method 1) for both *weighted_sentiment_diff* (no rolling mean) and *weighted_sentiment_rm_diff* (rolling mean). Rolling mean window length: 4.

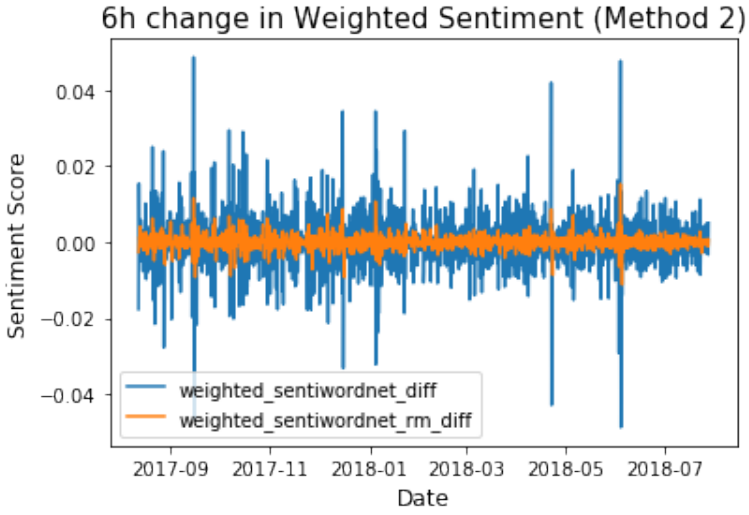

Figure 27: 6h aggregated change in Weighted Sentiment (Method 2) for both *weighted_sentiwordnet_diff* (no rolling mean) and *weighted_sentiwordnet_rm_diff* (rolling mean). Rolling mean window length: 4.

4.2 Lead-Lag Analysis

We would now like to know whether our calculated sentiment is in fact influencing the financial value of cryptocurrency or if it is the other way around. This question sounds rather straight forward, but research by Guo et al. (2017) on similar dynamics between sentiment and the stock market has shown a time-varying lead-lag structure between the two time-series. Traditional linear econometric models are therefore not suitable to study such a complex interaction over a large time-frame. For this reason, we will first conduct a cross-correlation analysis (cross-correlation is a measure of similarity among two series) between differently lagged time-series for a shortened time-frame of 1 month. This might give us a first indication on whether the lead-lag structure is constant or shifting over the span of the whole year. We will then employ the *Thermal Optimal Path* method developed by Sornette & Zhou (2005) to better visualize this lead-lag structure.

4.2.1 Lagged Cross-Correlation

The lagged cross-correlation Table 9 shows that when focusing on the year as a whole, there seems to be mainly a dynamic at hand where our sentiment score is leading and the Bitcoin value is lagging behind. A $p < 0.05$ denotes a correlation which is significant on the 5% level. Performing the same analysis over a rolling window of different size (from 1 month to 4 months) provides us with a plethora of different propositions though: Both leading and lagging dynamics for our sentiment score can be found, and they even lag by a different lag coefficient τ (See Table 10 and Table 11 as an example).
As expected, the lead-lag structure is not constant and we have to employ a more sophisticated analysis.

Table 9: Lagged cross-correlation between the differenced, 6h aggregated, weighted and rolling-meaned sentiment score from Method 2 (SentiWordNet) and the differenced BTC value for the whole year. p denotes the significance level value and *corr* the correlation coefficient. τ is the lag coefficient (so a τ of 1 means a lag by 6 hours).

Lagged Cross-Correlation		**BTC_diff**$(t-\tau)$		
		$\tau = 0$	$\tau = 1$	$\tau = 2$
weighted_ sentiwordnet_ rm_diff$(t-\tau)$	$\tau = 0$	p = 0.051 corr = 0.052	p = 0.066 corr = 0.049	p = 0.380 corr = 0.023
	$\tau = 1$	p = 0.031 corr = 0.057	p = 0.650 corr = -0.012	p = 0.980 corr = 0.001
	$\tau = 2$	p = 0.300 corr = 0.027	p = 0.780 corr = 0.007	p = 0.870 corr = 0.011

Table 10: Lagged cross-correlation between the differenced, 6h aggregated, weighted and rolling-meaned sentiment score from Method 2 (SentiWordNet) and the differenced BTC value for the time-frame 01.11.2017 - 31.12.2017. p denotes the significance level value and *corr* the correlation coefficient. τ is the lag coefficient (so a τ of 1 means a lag by 6 hours).

Lagged Cross-Correlation		**BTC_diff**$(t-\tau)$		
		$\tau = 0$	$\tau = 1$	$\tau = 2$
weighted_ sentiwordnet_ rm_diff$(t-\tau$	$\tau = 0$	p = 0.570 corr = 0.038	p = 0.008 corr = 0.180	p = 0.870 corr = -0.110
	$\tau = 1$	p = 0.330 corr = 0.066	p = 0.450 corr = -0.051	p = 0.990 corr = -0.001
	$\tau = 2$	p = 0.067 corr = 0.130	p = 0.750 corr = -0.022	p = 0.350 corr = 0.064

Table 11: Lagged cross-correlation between the differenced, 6h aggregated, weighted and rolling-meaned sentiment score from Method 2 (SentiWordNet) and the differenced BTC value for the time-frame 01.02.2018 - 31.03.2018. p denotes the significance level value and *corr* the correlation coefficient. τ is the lag coefficient (so a τ of 1 means a lag by 6 hours).

Lagged Cross-Correlation		**BTC_diff**$(t-\tau)$		
		$\tau = 0$	$\tau = 1$	$\tau = 2$
weighted_ sentiwordnet_ rm_diff$(t-\tau$	$\tau = 0$	p = 0.850 corr = 0.013	p = 0.900 corr = 0.008	p = 0.320 corr = 0.069
	$\tau = 1$	p = 0.390 corr = -0.060	p = 0.900 corr = 0.009	p = 0.980 corr = -0.001
	$\tau = 2$	p = 0.053 corr = 0.130	p = 0.620 corr = -0.034	p = 0.460 corr = -0.052

4.2.2 Thermal Optimal Path (TOP) Method

The name of the TOP method comes from statistical physics, where a very similar problem (*directed polymer in a quenched random potential landscape at non-zero temperature*) can be found. It can be split into four key ideas according to the publication in which it was first presented (Sornette & Zhou 2005):

1. A distance matrix of size $n * n$ is formed which allows to compare each value of time series $X(t_1)$ with each value of time series $Y(t_2)$ via the introduction of a distance $d(X(t_1), Y(t_2))$.

2. A one-to-one mapping $t_2 = \phi(t_1)$ is performed in a manner such that $X(t_1)$ and $Y(\phi(t_1))$ *match best.*

3. The introduction of a weighted average over many potential mappings removes non-informative noise in both time series and therefore eventually provides the optimal matching in step 2. This removal of non-informative noise can be tuned by changing the parameter T (Temperature).

4. The resulting mapping defines the lag between the two time series as a function of time and allows us to plot the time evolution of the relationship between the two time-series.

We employ the TOP method by using a modified version of an already existing Thermal Optimal Path software package by Amwatt (2019) in Figure 28 and 29 to look for the lead-lag relationship of Method 1 (Random Forest) and Method 2 (SentiWordNet) with the value of Bitcoin respectively. $\tau > 0$ implies the change in Bitcoin value to be leading and the change in sentiment lagging behind, while $\tau < 0$ suggests the opposite. We can now see why a single cross-correlation lag analysis over the whole year lacks any explanatory power: The lead-lag dynamic is indeed time-varying as the Bitcoin value is leading through the months of the cryptocurrency hype (See October 2017 until January 2018 in Figure 1) but lagging for the subsequent crash (See February 2018 until April 2018 in Figure 1).

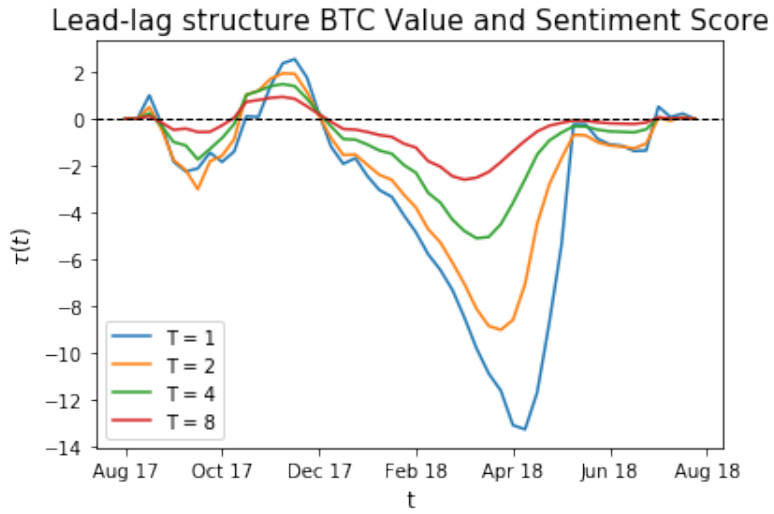

Figure 28: Thermal Optimal Path for different temperatures T between the differenced, 6h aggregated, weighted and rolling-meaned sentiment score from Method 1 (Random Forest) and the differenced BTC value. $\tau > 0$ implies the change in Bitcoin value to be leading while $\tau < 0$ suggests it to be lagging.

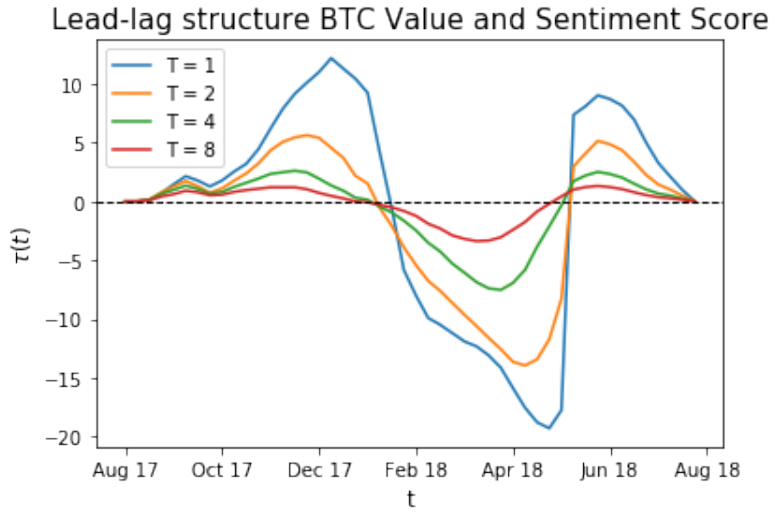

Figure 29: Thermal Optimal Path for different temperatures T between the differenced, 6h aggregated, weighted and rolling-meaned sentiment score from Method 2 (SentiWordNet) and the differenced BTC value. $\tau > 0$ implies the change in Bitcoin value to be leading while $\tau < 0$ suggests it to be lagging.

We can try to understand and interpret this dynamic by matching the lead-lag behavior with the period around the peak of the monetary value of Bitcoin: During the rise of Bitcoin, a phenomenon called *FOMO* (**F**ear **O**f **M**issing **O**ut, see Przybylski et al. (2013)) made people more aware and eager to invest into Bitcoin the further the BTC price rose (i.e. people *reacted* to a positive change for BTC). But at a certain point, the BTC hype started to dwindle when there were less and less people who are typically prone to FOMO which had not yet invested in BTC. And since less hype also means a lower sentiment score, this could have indeed marked the moment where the lead-lag dynamic switched and the negative sentiment started to impact the BTC price in a negative way.
This interpretation is only a hypothesis though and would have to be verified in future scientific work.

We will also have a look at the same dynamics for Etherium and Ripple. While Etherium shows similar dynamics to Bitcoin (Figure 30), in the case of Ripple (and many smaller altcoins not pictured here), the sentiment score seems to be leading for the majority of the time interval (Figure 31). One possible interpretation of this phenomenon is that smaller altcoins are more niche and only talked about by people with a certain expertise in the area of cryptocurrency and therefore less prone to be affected by hype and FOMO.

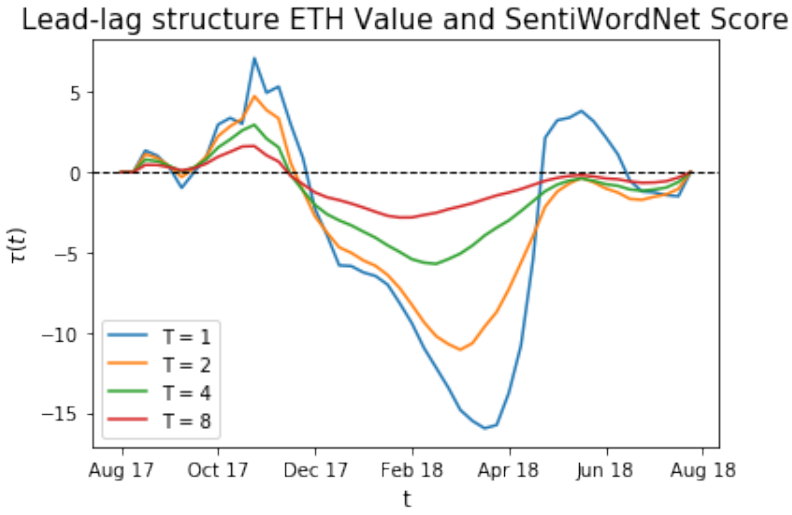

Figure 30: Thermal Optimal Path for different temperatures T between the differenced, 6h aggregated, weighted and rolling-meaned sentiment score from Method 1 (Random Forest) and the differenced ETH value. $\tau > 0$ implies the change in Etherium value to be leading while $\tau < 0$ suggests it to be lagging.

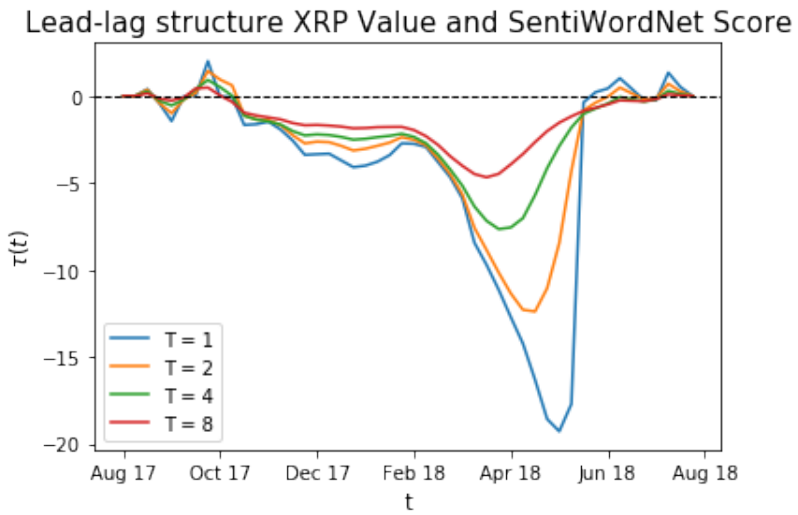

Figure 31: Thermal Optimal Path for different temperatures T between the differenced, 6h aggregated, weighted and rolling-meaned sentiment score from Method 1 (Random Forest) and the differenced XRP value. $\tau > 0$ implies the change in Ripple value to be leading while $\tau < 0$ suggests it to be lagging.

5 Conclusion and Outlook

The sentiment analysis part of the thesis presents multiple valid ways to extract a sentiment score from scraped tweets. Even a small labelled dataset provides enough information for a Random Forest model to achieve a high accuracy in a multi-class classification problem. And when leveraging labelled dictionaries like SentiWordNet, an acceptable accuracy can be achieved without a labelled dataset whatsoever.

The core question of this thesis is whether or not this sentiment score can be used to make predictions for the cryptocurrency market. While we didn't build a model to predict the market in real-time or to look for prediction accuracy over a certain time span (which we leave open for future research), we did look for correlation between the sentiment score and the market value of individual cryptocurrencies. Through our TOP analysis, we conclude that there indeed is a cross-correlation between the sentiment score and the market value after carefully preparing the two time-series by aggregating them and making them stationary. It is however a complex interaction with a shifting lead-lag structure. This implies an added difficulty for future research which might attempt to build a predictive model. Shortening the time-frame of the model could solve this issue, but it will lead to the loss of additional long-term information. Carefully balancing these advantages and disadvantages will be key for a successful predictive model.

A possible approach for such time-series models would be an autoregressive-moving-average (ARMA) model for the univariate case and a vector-autoregression-moving-average (VARMA) model for the multivariate case which could include more inputs like the amount of tweets in a certain time interval. But a machine-learning approach is possible aswell: A simple binary classification (*the value of coin x will rise* vs. *the value of coin x will fall*) could also provide interesting insights into the market. But no matter the approach, our TOP analysis shows that choosing an appropriate time-frame for the input data (or correcting for the shifting lead-lag dynamic) is crucial.

Bibliography

Amwatt (2019), 'thermal_optimal_path'.
URL: *https://github.com/amwatt/thermal_optimal_path/tree/master/thermal_optimal_path*

Baccianella, S., Esuli, A. & Sebastiani, F. (2010), Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, *in* 'in Proc. of LREC'.

Bollen, J., Mao, H. & Zeng, X.-J. (2010), 'Twitter mood predicts the stock market', *Journal of Computational Science* **2**.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984), *Classification and Regression Trees*, Wadsworth.

Ceriani, L. & Verme, P. (2012), 'The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini', *The Journal of Economic Inequality* **10**(3), 421–443.
URL: *https://doi.org/10.1007/s10888-011-9188-x*

Chollet, F. et al. (2015), 'Keras'.
URL: *https://github.com/fchollet/keras*

Colianni, S. G., Rosales, S. M. & Signorotti, M. (2015), Algorithmic trading of cryptocurrency based on twitter sentiment analysis.

Cryptocompare (2018), 'CryptoCompare.com'.
URL: https://www.cryptocompare.com// (visited Sept. 2018).

Dickey, D. A. & Fuller, W. A. (1979), 'Distribution of the estimators for autoregressive time series with a unit root', *Journal of the American Statistical Association* **74**(366), 427–431.
URL: *http://www.jstor.org/stable/2286348*

Farias, D. H. & Rosso, P. (2017), Chapter 7 - irony, sarcasm, and sentiment analysis, *in* F. A. Pozzi, E. Fersini, E. Messina & B. Liu, eds, 'Sentiment Analysis in Social Networks', Morgan Kaufmann, Boston, pp. 113 – 128.
URL: *http://www.sciencedirect.com/science/article/pii/B9780128044124000073*

Fellbaum, C. (1998), *WordNet: An Electronic Lexical Database*, Bradford Books.

Friedman, J. H. (2001), 'Greedy function approximation: A gradient boosting machine.', *Ann. Statist.* **29**(5), 1189–1232.
URL: *https://doi.org/10.1214/aos/1013203451*

Gayo-Avello, D. (2013), 'A meta-analysis of state-of-the-art electoral prediction from twitter data', *Social Science Computer Review* **31**(6), 649–679.
URL: *https://doi.org/10.1177/0894439313493979*

Guo, K., Sun, Y. & Qian, X. (2017), 'Can investor sentiment be used to predict the stock price? dynamic analysis based on china stock market', *Physica A: Statistical Mechanics and its Applications* **469**, 390 – 396.
URL: *http://www.sciencedirect.com/science/article/pii/S0378437116309384*

Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The Elements of Statistical Learning*, Springer Series in Statistics, Springer New York Inc., New York, NY, USA.

Hawkins, J. (2004), 'On intelligence (1st ed.)'.

Jefferson-Henrique (2018), 'Getoldtweets-python'.
URL: *https://github.com/Jefferson-Henrique/GetOldTweets-python*

Johnson, M. (2009), How the statistical revolution changes (computational) linguistics, *in* 'Proceedings of the EACL 2009 Workshop on the Interaction Between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?', ILCL '09, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 3–11.
URL: *http://dl.acm.org/citation.cfm?id=1642038.1642041*

Knerr, S., Personnaz, L. & Dreyfus, G. (1990), Single-layer learning revisited: a stepwise procedure for building and training a neural network, *in* F. F. Soulié & J. Hérault, eds, 'Neurocomputing', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 41–50.

Lampos, V. & Cristianini, N. (2010), Tracking the flu pandemic by monitoring the social web, *in* '2010 2nd International Workshop on Cognitive Information Processing', pp. 411–416.

Pennington, J., Socher, R. & Manning, C. D. (2014), Glove: Global vectors for word representation, *in* 'Empirical Methods in Natural Language Processing (EMNLP)', pp. 1532–1543.
URL: *http://www.aclweb.org/anthology/D14-1162*

Przybylski, A. K., Murayama, K., DeHaan, C. R. & Gladwell, V. (2013), 'Motivational, emotional, and behavioral correlates of fear of missing out', *Computers in Human Behavior* **29**(4), 1841 – 1848.
URL: *http://www.sciencedirect.com/science/article/pii/S0747563213000800*

Schank, R. C. & Tesler, L. (1969), A conceptual dependency parser for natural language, *in* 'Proceedings of the 1969 Conference on Computational Linguistics', COLING '69, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1–3.
URL: *https://doi.org/10.3115/990403.990405*

Sornette, D. & Zhou, W.-X. (2005), 'Non-parametric determination of real-time lag structure between two time series: the 'optimal thermal causal path' method', *Quantitative Finance* **5**(6), 577–591.
URL: *https://doi.org/10.1080/14697680500383763*

Twitter-FAQ (2019), 'Twitter Search API Offers'.
URL: https://developer.twitter.com/en/docs/tweets/search/overview/ (visited on 26.02.2019).

Vigna, P. & Casey, M. J. (2016), 'The age of cryptocurrency: How bitcoin and the blockchain are challenging the global economic order'.

WebHarvy (2019), 'Web Scraping'.
URL: https://www.webharvy.com/articles/what-is-web-scraping.html (visited on 26.02.2019).