VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Rotten Bitcoin Markets

Daniel Van Eijck

Supervisor: Kris Bubendorfer

Submitted in partial fulfilment of the requirements for
Bachelor of Science with Honours in Computer Science.

Abstract

This paper analyses cryptocurrency market manipulation tactics known as pump and dump schemes. Historic market data is scanned for patterns which indicate pump events and we explore how these pumps are organised through the social media platform Telegram. Logisitc Regression and Random Forest models are trained to classify these events with an accuracy of up to 88.9% with a precision score of 0.92. Next, the artifical models are trained to predict these events up to one hour before they occur, using specially constructed features extracted from candlestick and order book market data. Finally, we present initial work involving scraping social media for coin names and symobls, on the supercomputer Theta at the Argonne National Laboratory in Chicago, Illinois. We also explore how additional training features could be constructed by using sentiment analysis on the extracted social media data.

Contents

Figures

Chapter 1

Introduction

Bitcoin is a cryptocurrency that was created by an unknown person or group by the name of Satoshi Nakamoto in the year of 2009 [4]. Cryptocurrencies are digital currencies which only exist electronically and they are built with cryptographic tools which make transactions secure and hard to fake. One of the most important features of a cryptocurrency is that they are decentralized, meaning that there is no central authority or governor that controls how much of the currency is in circulation. Instead, the supply of Bitcoin is controlled by its design. Bitcoin is transferred between currency holders via a peer to peer network [14]. Each transaction that is made is stored and tracked in the "blockchain" which is a data structure based on encrypted Merkle trees [4]. These data structures are particularly useful for detecting fraud and false transactions, so if a single file in the chain is fraudulent, the blockchain will stop it from damaging the rest of the chain.

There are currently 17.6 million Bitcoins in circulation, with a limit of 21 million that can ever be created [4]. Because the currency is decentralized, transactions must be strictly verified in order to prevent people from performing fraud. The Bitcoin network is built on a "proof-of-work" system, where each block on the blockchain contains a hashcode [14]. Each transaction is verified by "miners" which use a computers' processing power to solve a computational problem which proves that the transaction is legitimate. Once a certain number of these problems have been solved, a new block is added to the blockchain. The hash of each block in the chain contains the hash of the previous block, which means that a block cannot be changed without redoing all of the work for the blocks after it [14]. Since Bitcoin mining is resource intensive but essential to the Bitcoin system, miners are compensated with Bitcoin rewards for each block that is added to the blockchain [4].

In 2017, Bitcoin's popularity rose significantly. This increase in popularity encouraged groups to make their own cryptocurrencies. Because Bitcoin is the main cryptocurrency and has the most coins in circulation, all other coins are considered "altcoins". The word is a combination of the words "alternative" and "coin" meaning that altcoins are every coin other than Bitcoin [6]. Many altcoins are based on the same framework provided by Bitcoin, that is a peer-to-peer network with a mining process which unlock new blocks to put onto the blockchain [6].

Because altcoins are not as popular as Bitcoin, they are mostly valued in terms of how much Bitcoin it costs for one of the altcoins. This is because altcoins are often not bought with USD currency, but rather traded for Bitcoin because it is the cryptocurrency with the highest market capitalization [1]. Market capitalization is the aggregate market value of coin represented in a dollar amount and it is calculated by multiplying the current market price

[1] `https://coinmarketcap.com/`

of a coin by the total number of coins in the market [17]. Due to Bitcoin having the highest market cap, it is the most widely accepted cryptocurrency as an actual form of payment.

In order to trade Bitcoin, or any other cryptocurrency, a person must create a Bitcoin wallet on a chosen exchange. Currently, the largest cryptocurrency exchange is Binance by trading volume[19]. There is a high amount of trading on the Binance exchange, which makes it easier for traders to turnover coins.

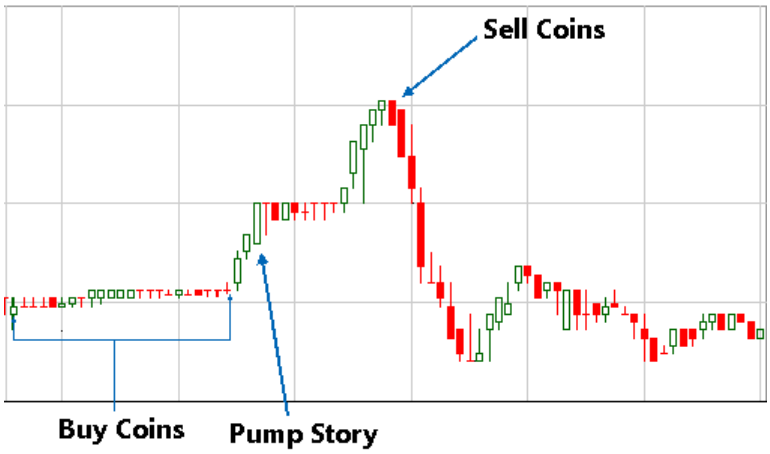
1.1 Problem

There is a big concern in the crypto community that some coin markets are turning "rotten", meaning that market prices are manipulated by groups of people and trading bots in order to turn a quick profit. Schemes called "pump and dumps" are often carried out on low valued altcoins, in which the price of the coin is pumped up by manipulators and later dumped when they sell all of their shares at the inflated price. Manipulators target altcoins with a low market cap because there is often not a lot of trading being done with those particular coins, which means that the price is consistently low and is easily pumped up with a sudden increase in trading volume.

Pump and dump schemes are a type of investment fraud where an individual or group of people buy a large number of shares at a low price which results in the share value rising rapidly (the pump). The group then convinces other people to buy these shares at the increased price, by spreading misleading positive information about the stock. Once all of the shares have been sold, the share value returns to the original price or even lower (the dump), leaving shareholders with shares that are worth less than they bought them for. The people who organise pump and dump events have the potential to make a lot of profit because they create an oppurtunity to sell a large number of shares at an inflated price. Previous research in "Stock market manipulation-theory and evidence" [15] has shown that investors who have insider information of the company have a high possibility to be a manipulator. Market manipulation tactics such as pump and dumps are illegal [1] and there has even been offers by the Commodity Futures Trading Commission of monetary rewards for pump and dump whistleblowers.

Pump and dump events can be identified by looking at the historical price data of crypto coins. A common pattern is that the pump organizers will start to accumulate shares while the market price is still fairly low. Once they start accumulating the shares, this drives the price up. As the price is rising, the pump organizers may start spreading false positive information about the coin, creating some hype. This is where it is easy to suck in novice buyers, because they believe that the price will keep going up. There are also a large number of naive trading bots designed to buy coins when their price suddenly begins to increase, which can contribute to inflating the price as well. Once the pump organizers are satisfied with the inflated market price, they sell all of their accumulated shares at once, making a big profit. This sends the market price down and leaves all of the buyers that were tricked by the pump with shares that are worth much less than they paid for them. Figure 1.1 shows an example of the typical shape of a pump and dump event.

In order to pump a coin, a large amount of buy orders must be completed in a very short amount of time. Therefore, groups which perform pumps need to have a significant number of members in order for the pump to be successful [12]. Anonymous messaging platforms such as Telegram are used to plan and execute pump and dumps on alt coins. It is common to find pump and dump communities on Telegram where pumps are advertised. The admin will post messages in the channel promoting the time and day of the pump [12].



Source: https://restislaw.com/cftc-crypto-pump-dump-whistleblower

Figure 1.1: Structure of a pump and dump

The admin also posts the rules of the pump to make sure all of the participants know what to do when the pump starts. Participants are told to buy big when the event starts and then to sell in small chunks in order to keep the price high. At the time of the pump, the admin posts a text message or picture which indicates what coin is to be pumped. Everyone then rapidly begins to buy the coin to try and get in before the price skyrockets. Results from the paper "Collective Sensemaking in Cryptocurrency Discussions" [9] concluded that there is commonly two types of cryptocurrency users. There are people that are concerned with the development of crypto technology and these people view the coins as a legitimate form of currency [9]. The other type of user involves people who see cryptocurrencies as purely an investment opportunity and these are the people that are more prone to act on market hype. The latter types of users are prone to falling for pump and dump schemes, as seen in research done by Dirk G. Baur and Thomas Dimpfl in "Asymmetric volatility in cryptocurrencies" [2]. This research showed that there is a large portion of crypto users that buy coins after seeing positive news about the assets, in fear of missing out of potential profits when the price rises.

The individuals who organize pump and dumps are the ones who profit the most [20]. This is because only the admin of the group knows what coin is going to be pumped. This means that during the days leading up to the pump, they can slowly accumulate coins that they can sell at the inflated price. There are two main types of social channels that pump details get announced on: premium (private) and public channels. In order to part of a premium channel, members must pay the channel admin a subscription fee in Bitcoin. It is common for pump channel admins to release the coin to pump to the premium channel up to 5 minutes before the public channel [12], so that the premium members have a chance to buy in at a lower price. When pump participants start rapidly buying coins, it is in fact the admin selling everyone their accumulated assets. When the price peaks, everyone attempts to sell their coin, however they often struggle because there are no buyers. In the end, the admin and members in the premium channel make most of the profit [12]. The admin then reports how successful the pump was, by posting their percentage of profit in the public channel. This can be anywhere from 100% - 500% and makes future pumps look attractive to new members of the community. This also encourages newcomers to pay for the premium channel, because the admin is promoting how much more profit can be made when the coin to be pumped is known early [12].

1.2 Proposed Solution

This project will investigate the pattern of pump and dump schemes and attempt to build more predictive tools that can identify pump and dump events before the affected market is dumped. Several forms of historical market data are downloaded and analyzed, including candlestick and order book data. Candlestick data is composed of the open, close, high, low prices and trading volume for a selected trading time interval. Order book data contains information about all of the filled orders for a trading interval. Order book data includes the price at which the order was filled, the volume of currency being traded and Boolean values which indicate whether the buyer placed the order (rather than the seller) and whether the price at which the order was filled was the best match. The final deliverables include python scripts for:

1. Downloading candlestick data and order book history from the Binance exchange
2. Extracting sections of pumped and non-pumped market history from the candlestick data using a peak detection algorithm
3. Scraping Telegram and Reddit for coin mentions
4. Verifying pumped / non-pumped data points by cross referencing Telegram coin mention dates
5. Training artificial models to perform binary classification on data points
6. Merging candlestick data and order book data for each data point
7. Constructing features from candlestick and order book data
8. Training artificial models to predict pump events with constructed features
9. Basic sentiment analysis on Reddit data

1.3 Evaluation Method

The solution is evaluated by training the artificial prediction models and observing the precision at which pumped data points can be classified. A precision score of less than 0.8 or 80% will be considered as failure in terms of being able to accurately predict pump and dump events. This is due to the fact that the purpose of the solution is to protect traders from losing money on risky coin investments. If the model cannot provide accurate predictions, then this may cause the trader to lose money. Scraping coin data from Reddit is performed on the supercomputer Theta of the Argonne National Laboratory in Chicago. The Reddit data is evaluated by performing some initial sentiment analysis experiments.

1.4 Contributions

Section 2 of the report summarizes previous published research performed on pump and dump schemes in altcoins. This section discusses the process followed by previous researchers as well as the results from many different classification and prediction techniques. Both the results of previous researchers and the results presented in this paper provide solid evidence that pump and dump manipulations occur in cryptocurrency markets frequently, which results in some novice buyers losing money.

Section 3 of the report explains the key design decisions made when undergoing the research, many of which were motivated by the results of previous published research. First, the choice of dataset, trading exchange and selected coin set is defined in respect to limitations presented by large availability of crypto market data. Section 3.2 carries out testing of peak detection results to identify the most appropriate market data metric to identify pump and dump events with. Section 3.3 describes a technique used for cross referencing peak detection data points with discussion of crypto assets in social media channels. Section 3.4 discusses and disproves a hypothesis drawn from previous research in respect to pump events normally happening on the half or full hour. Section 3.5 describes the techniques used to build an up-to-date list of crypto related subreddits and then explains the design decisions when constructing the Reddit scraping script.

Section 4 contains the details of the python implementation for each stage of the research. In section 4.1 we define how the peak detection algorithm is implemented and what its output looks like. Section 4.2 describes the implementation details of the Telegram scraping script used for collecting coin mentions on public pump and dump channels and how these coin mentions are used to verify the data points. In section 4.3 we briefly introduce the process we used when performing binary classification using various artificial models. This section also includes a breakdown of the constructed features used when attempting to perform pump predictions.

Section 5 contains an evaluation of the results gathered from the produced Reddit scraping script [A.0.5]. In section 5.1 we present the market data points that were collected, the Telegram coin mentions and the results of validating the data. In section 5.2 we present the successful binary classification results using Logistic Regression and Random Forest models. Section 5.3 presents the classification results achieved when feeding only a limited amount of data to the models in an attempt to classify the data points without the peak in trading volume being shown to the models. Section 5.4 presents the prediction results when using the constructed features shown in Section 4.3 on both the verified and non-verified datasets.

In the final section we describe various plans for future work which includes: adding data from more exchanges, using data from more social platforms to verify the data, performing stricter data verification and performing real time data gathering. This research has been focused on identifying and predicting very short-term pump events, however in section 6.5 plans are made to use social media sentiment analysis data in pair with coin market data to perform prediction of more longer-term pumping schemes.

Chapter 2

Background

There has been a lot of recently published research concerning the manipulation of cryptocurrency markets. The paper "Adaptive Hidden Markov Model with Anomaly States for Price Manipulation Detection" [3], published in 2015, explores detecting currency manipulations with an adaptive hidden Markov model with anomaly states in regular stock markets. A similar paper "Stock Price Manipulation Detection Based on Mathematical Models" [11], published in 2016, describes how stock market manipulations can be detected based on mathematical models which define the shape of market data effected by pump and dump schemes and spoof trading. A paper published in 2017, called "Cryptocurrency Pumping Predictions: A Novel Approach to Identifying Pump and Dump Schemes" [16] extends from this work and explores predicting pump and dumps with Support Vector machine models and Neural Networks. One of the most recent papers "The Anatomy of a Cryptocurrency Pump-and-Dump Scheme" [21], the most recent version being published in 2018, focuses identifying and predicting pump and dump schemes carried out in altcoin markets on various exchanges.

While this research is mainly based on attempts to classify and predict pump and dump events based on historical market data, there is also some potential to incorporate social media data as way to construct additional features for manipulation prediction. A paper published in February 2019 called "Identifying and Analyzing Cryptocurrency Manipulations in Social Media" [13] analyzes the social media activity involved in pump and dump schemes across platforms such as Twitter and Telegram. This work will be extremely useful for the future work of this research, as we incorporate social media data into the prediction models. The most recent research on cryptocurrency manipulation is a paper called "Cryptocurrency Pump-and-Dump Schemes" [18] which is still in its second draft phase.

In this chapter, we will focus on summarizing the design and results of the 3 most relevant papers to this research. First, "The Anatomy of a Cryptocurrency Pump-and-Dump Scheme" [21] is summarized, as this paper contains a lot of useful information on what pump and dumps look like across multiple exchanges and discusses techniques for extracting features from historical market data for training prediction models. Next, "Stock Price Manipulation Detection Based on Mathematical Models" [11] is summarized because this paper gives excellent insight to the properties pump and dumps exhibit in historical market data. Finally, "Cryptocurrency Pumping Predictions: A Novel Approach to Identifying Pump and Dump Schemes" [16] is summarized as it contains successful methods to training models to predict pump and dump events.

2.1 Anatomy of a Pump and Dump

The most recent published study into the coordination of pump and dump schemes was carried out in November 2018 by Jiahua Xu and Benjamin Livshits from the University of St. Gallen Imperial College London [21]. The study goes into detail about the anatomy of a cryptocurrency pump and dump scheme by analyzing a pump and dump case study. The paper also investigates patterns which arise in historical data of pump and dumps by analyzing 220 pump and dump activities that were coordinated in Telegram groups from July 21, 2018 to November 18, 2018. Finally, the study builds and evaluates a predictive model that can predict the likelihood of a coin being pumped 1 hour before the pump begins.

In order to gather pump data, the study referred to over 300 Telegram groups that are dedicated to conducting pump and dumps. The Telegram groups were on a website [1] that is dedicated to promoting the most active pump and dump Telegram groups. From reading through historic chat data in these pump groups, there is a clear pattern that demonstrates how the pump events are organized. First, the pump organizer will send out a pre-pump announcement that discloses the exact time and date for the pump event, as well as the targeted exchange. In the hours leading up to the pump, the organizer will constantly post reminders for how many minutes are left until the event begins. The organizers will also post the rules of the pump, which are usually something like:

1. Buy the chosen coin really fast
2. Promote the coin on social media
3. Hold the purchased coin for at least a few minutes to give time for more buyers to join the pump
4. Sell in small pieces
5. Only sell at profit, that is, only sell the coin above the current price

When the time comes for the pump event, the organizer will post the coin to pump in the form of text in a message, or as an image into the public channel [21]. In some cases, non-machine-readable images are used to prevent Telegram scraping bots from identifying the coin and interfering with the pump. The price of the coin will quickly skyrocket as pump participators rapidly buy the coin, until the price peaks. A few minutes after the pump has started, pump participants begin to sell at a profit which causes the price to fall rapidly. As soon as the dump starts, pump participants panic-sell their assets because they are worried the coin price will drop and they will lose the potential to make profit [21]. Once the price of the coin has returned close to the starting price before the pump, this is an indication that the dump is over because most investors would rather hold on to the coin in hopes that it will be pumped again in the future.

The paper [21] presents a pump and dump case study. The BVB coin was pumped on November 14, 2018 on the Cryptopia exchange. The BVB coin was rated 1/5 stars on Cryptopia, which means that the coin had very little recent activity before the pump. The study hypothesizes that this was a deliberate choice, as the projects associated with the coin were not active. Therefore, the coin could not resist pump and dump activity because it did not have a stable market beforehand. In this case, a stable market refers to a market with regular trading happening, however the BVB coin had very little trading happening at all. One of

[1] `PumpOlymp.com`

the most significant features that was observed from this pump was the sudden increase in trading volume. Before the pump, the coin had a trading volume of close to 0 coins. However, during the first 15 minutes of the pump event, BVB's trading volume peaked at 1.41 BTC. It was observed that the pump event induced a fake demand for the coin as indicated by the increase in buy volume. Xu and Livshits also observed that the total buy volume outweighed the total sell volume for the coin during the pump period. This indicates that some pump participants bought coins during the pump stage but were not able to sell these coins at the inflated price before the dumping stage.

The study [21] made some interesting observations about the organization and use of Telegram channels to coordinate pump and dumps. Using the website Pumpolymp and searching for channels manually, Xu and Livshits were able to form a list of 358 Telegram channels used for conducting pumps. Of those channels, 43 had been deleted from Telegram. From the channels that were still online, 168 had not been active for at least a month. The researchers hypothesized that channel admins may be very cautious with leaving behind evidence of pump activity, so they delete pump messages and channels. This leads to another hypothesis, that pump and dump schemes have a "hit and run" characteristic. After a few times participating in pumps and not earning any profit will make pump participants lose interest in participating. This results in lower numbers in pump participants as the lifetime of the pump channel grows. This could cause pump organizers to periodically delete the pump channels and create new ones in the hope of attracting fresh inexperienced participants.

The paper [21] then describes how they collected historic pump and dump events with the Pumpolymp website. The website has a page which lists historic pumps limited to the last 3 months. The website lists the coin, the exchange where the coin was pumped and a count of how many times the coin has been pumped in the last 3 months. The page also displays the dates and times that each historic pump occurred, as well as the Telegram group that organized it. In order to collect the data, the Xu and Livshits scraped the historic pump data from this web page many times over the course of several months. The researchers decided to discard any of the listed pumps that did not occur extremely close to the full hour or half hour, since "an organizer would normally not choose a random time for a pump-and-dump" [21]. This is because times that are on the full or half hour are easier for participants to remember, which means more people to remember to participate in the pump. The researchers also decided to discard any events where the trading volume or price for the coin did not significantly increase, because these are two clear indications of a pump. Once these filters were applied, the researchers went through and checked each example manually in the Telegram channels. A data point was only considered valid if the Telegram messages were in a style of a coin announcement, which was defined earlier in the paper. The result was a total of 236-coin announcements from July 21, 2018 to November 18, 2018. If two data points had the same coin and happened on the same exchange within 3 minutes of each other, the researches considered this as a single data point. After removing duplicate data points, the result was a total of 220 unique pump and dump events.

An analysis on the 220 pump and dump events was performed. They firstly calculated the distribution of pumps that occur on each exchange. The exchange with the most pump events was Cryptopia with 148 (67%) followed by Yobit (18%), Binance (11%) and finally Bittrex (4%). Next, the researchers analyzed the average increase in trading volume on each exchange. The researchers took the average trading volume before pumps and the average trading volume during a pump in order to find the difference. Cryptopia had the most significant increase of trading volume during pumps with an increase of 7105%, followed by Yobit with an increase of 2900%, Binance with an increase of 1318% and finally Bittrex

with an increase of 834% [21].

The study [21] draws some conclusions as to what role exchanges have in pumping schemes. The large exchanges such as Binance and Bittrex have a much larger user base than Yobit and Cryptopia, which means that abnormal price hikes quickly attract a large number of users to the pumping coin. The smaller exchanges such as Yobit and Cryptopia tend to host smaller start up coins with very low liquidity, meaning that the prices of these coins are more vulnerable to manipulation caused by sudden increase in trading volume. Due to these factors, it is observed that pumps which occur in Yobit and Cryptopia result in a much higher price increase percentage than Binance and Bittrex.

Xu and Livshits made an interesting discovery when analyzing the relationship between the number of message views the pump announcement received and the price percentage increase caused by the pump. They found that there was a negative correlation (-0.162) between the message view and pump gain. Because of this, it is hypothesized that bots are used to read the text-based pump messages and act on the signals. Bots do not require membership of the group, therefore a bot reading the message will not increase the number of message views on Telegram.

Xu and Livshits analyzed the market caps for each pumped coin to try and find any patterns. It was observed that pumped coins usually have a market cap that is under half of Bitcoins market cap of 1.74×10^7. On the exchange with the most pump activity, Cryptopia, a majority of the pumped coins had a market cap of under 100 BTC. They also investigated patterns in the hourly log return of each coin ranging from 48 hours before and 3 hours after the pump event. The researchers found interesting results in the hourly return data for coins on Cryptopia, where in multiple instances the hourly return one hour before the pump exceeded the hourly return during the pump. The study explains that this behavior could be caused by pump organizers utilizing their insider information to buy the coin in the hours leading up to the coin announcement.

The study [21] then presents two models that attempt to predict whether or not a coin will be pumped based on coin features and market movements. Xu and Livshits decided to focus on the Cryptopia exchange because that is where the most pump and dump activity was observed. Previous analysis of market movement prior to a pump event indicated that there could be signs of pump organizers buying the coin before it is pumped. Therefore, the researchers placed great emphasis on coin features related to market movements, such as coin price and hourly returns.

The coin data used was split into training, validation and test sets. The training set consisted of 27,759 data points (58.3%) and contained 78 true pumped instances. The validation set consisted of 10,106 data points (21.2%) and contained 28 true pumped instances. The test set had 9,755 data points (20.5%) and contained 27 true pumped cases. All of the data sets contained data from different time periods. The training set covers July 21 to October 10, the validation set covers October 10 to October 29 and finally the test set covers October 29 to November 18.

The first model the researchers made was a classification model using Random Forest (RF) with stratified sampling. 3 random forest models were tested, each with ranging sample sizes and number of trees. The researchers applied the RF models on the dataset and then performed some evaluation on which coin features had the most impact on the output classification. By analyzing the mean decrease in Gini coefficients with the RF models, the Xu and Livshits made the following observations: The two most important coin features were the coin market cap (measured when no pump activity was observed in the Telegram channels) and the 1-hour log return before the pump. Market movement features that are

recorded very shortly before the pump (1-hour return and 1-hour volume) are more significant than features that describe longer term movements. The return features are generally more significant than volume features. Features that are specific to the exchange are least important.

The second model the Xu and Livshits made was a Logit Regression model using Generalized Linear Model (GLM). Because the coin price distribution and market cap distribution data were heavily skewed, the researchers also applied a least absolute shrinkage and selection operator (LASSO) in order to produce some regularization to the GLM models. Just as with the RF models, the researchers made three GLM models all with different shrinkage parameter values. When evaluating the importance of coin features on the outputs of the models, the researchers observed some things that align with the results of the RF models: 1-hour return seems to be the best indicator of whether the coin will be pumped. They also found that the higher the 1-hour return, the more likely the coin will be pumped. Similarly, the more times a coin has been pumped in the past, the more likely it will be pumped again.

Both models were capable of predicting whether a coin will be pumped with a probability between 0 and 1. A thresholding algorithm is then applied to get a binary true or false output. Xu and Livshits compared both models by testing the models at different threshold values and producing receiver operating characteristic graphs. The RF models are superior to the GLM models, because in both the training and validation data sets, the RF models have a higher area under curve (AUC). The best RF model had an AUC of 0.9320 in the validation data set while the best GLM model only had an AUC of 0.8631. While both models had decent performance in the training set (AUC $>$0.9), the GLM model performs poorly on the validation set which is an indication of over fitting.

2.2 Manipulation Detection

In June 2016, a paper called "Stock Price Manipulation Detection Based on Mathematical Models" was presented in the International Journal of Trade, Economics and Finance [11]. The paper was written by Teema Leangarun, Poj Tangamchit, and Suttipong Thajchayapong. The paper investigates two popular methods of manipulating the stock markets: pump and dump schemes and spoof trading. Mathematical definitions for these manipulation methods are presented and a feed forward neural network is designed which achieved an accuracy of 88.28% when detecting pump and dump events.

The ability to be able to detect pump and dump events effectively depends on the amount of data historical data available. The paper [11] discusses two types of market data: level 1 and level 2 data. Level 1 data is made up of successfully completed buy and sell orders for a stock and has the format of open, high, low, close price and volume (OHLCV) at different time frames. Level 2 data consists of all of the level 1 data as well as buy and sell order that are cancelled. Generally, level 2 data is only available to market authorities and only level 1 data is available to the public. Therefore, the researchers attempted to make a neural network model for detecting pump and dump events using only level 1 market data.

The paper [11] discusses the concept of spoof trading. Spoof trading is when a market manipulator places a large amount of passive buy orders at a higher price than the current market price, in order to trick other traders into thinking the stock should be sold at that price. Once the manipulator has made enough profit by selling their stocks at the inflated price, they cancel their buy orders and the market price returns to normal. The main difference between spoof trading and pump and dumps is that the manipulator enters many passive buy orders that they have no intention of being filled, whereas in pump and dumps

the participants enter many buy orders that need to be filled in order for the price to inflate. Level 2 data is required for identifying spoof trading, because this data contains market information at a sufficient depth that includes all orders (matched or unmatched) and an identification number which indicates the individual placing the orders. It is useful to have this ID number because then you can identify all of the orders placed by the manipulator.

The paper [11] defines a mathematical model for identifying pump and dump events. They define 3 conditions that must be meet for a pump and dump event to be valid: 1 pump condition and 2 dump conditions. Firstly, the dump conditions are checked and if they hold, the pump condition is checked. The first dump condition is as follows: the amount of canceled buy orders is more than 50% of the average volume of matched buy orders. The second dump condition is that the difference between the highest price of sell orders and the lowest price of sell orders is more than 15%. If both of these conditions hold true, then the pumping condition is checked: the difference between the highest matched buy order and the lowest unmatched buy order at the starting period is greater than 0.2%. This is the model which the researchers used to acquire training pump data for the neural network.

The neural network that was developed was made up of an input layer with 25 nodes, a hidden layer with 3 nodes and an output layer with 1 node. The network simply outputs a 1 if the data point is in fact manipulated or 0 if it is not. The network was trained with only level 1 data using a supervised training method with back propagation. The researchers chose to use 1-minute interval data from stock markets including Amazon, Intel and Microsoft. The data was split into two sets: training data for training the network and a test data set for evaluating the networks performance. The model was trained and tested with 22 sets of data with approximately 50% manipulated data points and 50% non-manipulated data points. The model was evaluated by leave-one-out cross validation and the performance evaluation was based on the mean square errors. Results showed that the neural network could successfully detect pump and dump cases with a mean square error of 0.0641. However, the model could not detect spoof trading when trained on only level 1 data, which can be seen in the mean square error of 1.3992.

2.3 Machine Learning

At the end of 2017, students from the Department of Computer Science, Stanford University presented a model which can identify pump and dumps using historical market data [16]. The students obtained market data from the exchange Poloniex using the publicly available API.

Initially, when the students were collecting pump data, they were collecting it manually by checking price data and extracting the obvious pumps. However, in order to collect a large enough data set to train a model, this approach took too long. Instead, they collected 24-hour windows of data where the coin went up 15% in price compared to Bitcoin and then collected the relevant order book data for that time period. Because each 24-hour block of data contained a different amount of orders, the students utilized Bresenhams's line algorithm to split the data points into N evenly spaced intervals that they could average the order data between.

The students implemented and tested two types of machine learning models. The first model that the students tested was a Support Vector Machine using the python sklearn module. The model was evaluated by training the model using a randomized half set of the whole data set and then testing the classification accuracy using the other half of the data set. This was performed on loop 16 times and the results were averaged. The students tested the model using increasing values of N (used for splitting and averaging the order book data). The best results were produced with N = 2600, with the model successfully identifying pump and dumps 94.6% of the time in the training data and 78.13% of the time in the test data.

For the second model, they took a clue from the paper "Stock Price Manipulation Detection Based on Mathematical Models" [11] and implemented a feed forward neural network. This model was tested in the same way as stated previously, trained on a randomized half of the data set and tested on the other with increasing values of N. The model performed best with N = 2300 and was able to successfully classify pumps 82.8% of the time with the training data and 81.2% of the time with the testing data. This is a substantial improvement from the SVM model, especially because the accuracy on the training data and testing data are nearly equal, showing that the neural network is less prone to over fitting to the training data than the SVM.

Finally, the students implemented a second neural network model that was only trained on the first 12 hours of data, to see whether it could predict whether a pump and dump would occur in the second 12 hours of data. The results were extremely promising with N = 3400, with the model achieving 78.2% accuracy on the training data and being able to predict a pump in the testing data 82.5% of the time.

The students mention that if giving more time to improve the project, they would attempt to add order cancellation data (level 2 data) to the data features so that they could identify spoof trading, which is a good indicator that a pump and dump event is occurring.

Chapter 3

Design

3.1 Choice of dataset, exchange and coins

The first step to identifying historical pump and dump events is to pick a trading exchange and form a method for retrieving historic trading data. Binance is one of the largest exchanges to date [19] and has a supplied API which returns coin information in json format. The project python-binance[1] is an unofficial python wrapper for the official Binance API and supplies methods for retrieving the candlestick data for a particular coin within a given date range. You must also supply the interval for the data you want to retrieve in the form of 1, 3, 5, 15 and 30-minute intervals or larger intervals such as 2, 4, 6 and 8 hours. Candlestick data returned for a coin includes: open time, open price, close price, highest price, lowest price, close price, trading volume and close time. These values are defined in more detail below:

1. Open time – the time in milliseconds indicating the start of the trading period
2. Open – the price of the coin at the start of the trading period
3. Close price – the price of the coin at the end of the trading period
4. High – The highest price of the coin between the open time and close time of the trading period
5. Low - The lowest price of the coin between the open time and close time of the trading period
6. Close – the price of the coin at the end of the trading period
7. Volume – the amount of coin that has been traded within the trading period
8. Close time -the time in milliseconds indicating the end of the trading period

Special care was taken when deciding what interval range the data was retrieved in. Because organized pump and dump schemes often happen in a short amount of time [21], it is appropriate to use a shorter interval such as 15 or 30 minutes rather than longer time spans such as 6 hours or 1 day. Some initial testing was done with 6 months of data from the coin NEBL where 1 pump was detected using a 6-hour interval and 9 pumps were detected using a 30-minute interval. We can see that many more data points can be detected using the 30-minute interval data, therefore this interval was used for the initial analysis of the

[1] `https://github.com/sammchardy/python-binance`

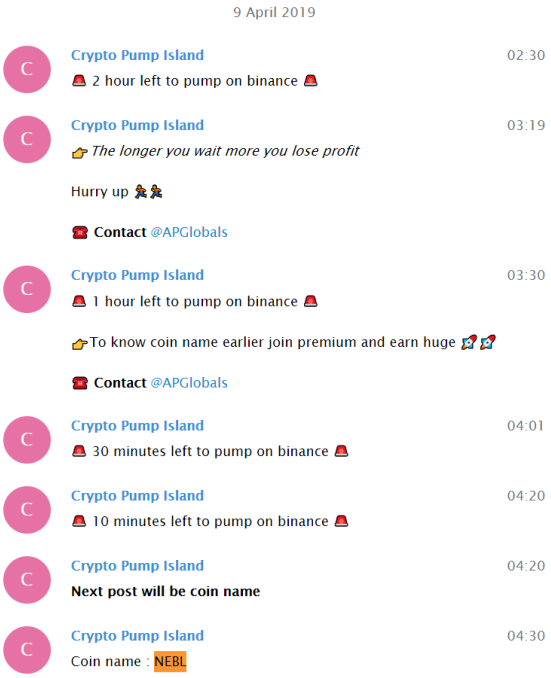
market data. The next step was to identify a list of coins to perform data collection on. A list of alt-coins paired with BTC listed on the Binance exchange was scraped from the CoinMarketCap website on the 1st of June 2019. The result is a list of 147 coins that were used for initial data collection.

3.2 Choice of peak detection metric

Once the market data was retrieved, a peak detection algorithm was used in order to identify abnormal snapshots of market movement where the price of the asset increased significantly in short period of time. The chosen algorithm for this task was the smoothed z-score peak detection algorithm. This algorithm works by calculating the moving mean for a given widow size on the time series data and if the next data point is so many standard deviations away from this moving mean (determined by a threshold value) then the algorithm will output a signal. If the data point is a threshold number of standard deviations above the moving mean, the algorithm will signal a 1, otherwise if the data point is a threshold number of standard deviations below the moving mean, the algorithm will output -1. If the data point is within the moving mean, the algorithm will output 0. The sensitivity of the smoothed z-score algorithm can be tuned using 3 parameters: lag, threshold and influence. The lag parameter determines the size of the sliding window when calculating the moving mean. The threshold parameter determines how many standard deviations a data point must be to be considered a signal. The influence parameter determines how much a signal data point effects the threshold for future data points. For example, an influence of 0 means that future signals are detected based on a threshold that is not affected by past signals. An influence of 0 is therefore the most robust option, because we will assume that pumps happen individually from each other.

The next key design decision involved determining which market feature should be scanned for peaks. Three potential candidates are the close price, highest price and trading volume. The closing price of the asset is a good candidate because we would expect it to be artificially inflated during pump events. Similarly, a peak in the highest price would be a good indicator of artificially inflated prices, since this market feature records the highest price of an asset during a trading period. Finally, results seen in [21] show that trading volume is a good candidate, as it can jump to 1318% higher than normal during pump events on the Binance exchange. In order to select the best market feature to run peak detection on, several tests were performed to gain insight on how much noise each feature produced. Before any results could be compared, it was necessary to tune the peak detection parameters for each market feature. In order to tune the threshold, we used the coin NEBL whose data contains multiple obvious pumps in the month of April 2019, with matching Telegram coin announcements in the Crypto Pump Island channel. An example of this pump and their Telegram messages are shown in Figure 3.1. We performed several tests with different threshold values to see how many standard deviations away from the 12-hour moving mean was optimal for identifying the obvious pumps. For each threshold value tested, we checked the output graphs manually to see how many data points looked like valid pumps and how many were invalid. We then lowered the threshold if there were not many invalid data points or raised the threshold if too many data points looked like normal market movement and not pump events. We observed that peaks in volume were much more significant than peaks in price, meaning that there is a bigger difference between the 12-hour moving mean and peaks in volume than there is with closing / highest price. As a result, we found that the optimal threshold for trading volume is 40 standard deviations from the moving mean, while only 14 standard deviations was optimal for the price values.

(a) NEBL coin pump on Binance



(b) Telegram coin announcement message

Figure 3.1: NEBL pump and dump

Indications given from paper [21] suggest that organized pump events usually occur on the full hour or half hour. Therefore, we will assume that the highest peak in the data is only a valid pump signal if it occurs within 5 minutes from after the full hour or half hour. If we filter the potential pumps according to this condition, then we end up with a final pump list. In this case, noise is considered as the percentage of potential pumps that are filtered out due to the time restriction. I hypothesize that the volume market feature will produce the least noisy data. This is because price fluctuations in crypto assets are extremely common during everyday trading, however increases in trading volume indicate more people are trading a specific coin than usual, which is a key indication of a pump event. Below are the test results for 3 market features across 147 coins on Binance between 1st of January and 1st of June 2019 (0.5 years):

Market feature	Total detected pumps	Valid pumps	Pumps filtered out (Noise)
Volume	1700	414	75.6470588235294%
High	1798	473	73.69299221357063%
Close	547	102	81.35283363802559%

We can see that the high price market feature produced the greatest number of valid pumps at 473 and it is the least noisy data with 73.69% of data points being filtered out. Initially, this makes sense because pump events cause peaks in the highest price of a coin during a trading period. However, this result does not support the hypothesis. Upon inspection of the output data, there are several coins that produced over one hundred potential pumps with many of them being filtered out. When inspecting the graphs for these data points, it is obvious that they are invalid indications of pump events. This could be because coins with extremely low trading activity can trigger the peak detection algorithm when some activity does take place. For example, the coin NPXS produced 765 potential pumps and

260 pumps once filtered using the time restriction. Of the 260 pumps, none of the graphs resemble pump events, therefore we decided to eliminate any coin which produced over 100 data points because they do not contain any valid data. As a result, the following coins were removed from the list: BTT, NPXS and SC. New results were produced with the new coin list, now 144 coins long:

Market feature	Total detected pumps	Valid pumps	Pumps filtered out (Noise)
Volume	1549	388	74.95158166559071%
High	553	116	79.02350813743219%
Close	547	102	81.35283363802559%

The above results now align with our hypothesis. Due to trading volume producing a significantly higher number of data points with the least amount of noise, it was selected as the market feature for further testing.

3.3 Verifying data using telegram

The next key design decision was to attempt to validate each data point by cross referencing the pumped coin and pump date with coin mentions on Telegram. As seen in paper [12], Telegram is a social media platform where pump organizers schedule and conduct pump events within big group channels composed of potential pump participants. At the time of the pump, the channel admin will send out a message containing the chosen coin to pump. Many of the pump channels are run by bots, which means that the coin announcement message is in the same format each time. This makes it easy to scrape coin announcement messages by using regex patterns. On June 14th, a list of public pump and dump Telegram channels was collected from the PumpOlymp website. Each channel was opened in Telegram and if the channel still existed, the data was exported in the form of HTML files using the Telegram desktop chat history export tool. This resulted in 167 unique pump channels [A.0.2] with their history being exported.

3.4 Filtering data points

Initial testing for noise on the market data involved filtering the potential pumped data points depending on whether the peak trading volume of the asset occurred within 5 minutes of the full or half hour. Upon reading the results from the Telegram coin mention scrape, it became apparent that many of the organized pumps rely on announcing coins on separate pump channels in staggered times. For example, an extremely common pattern seen in public pump and dump channels is that the admin will post a coin to pump which has already been disclosed in premium (private) pump groups many minutes prior. In this way, members of the private pump groups have a better chance at buying the asset low and then selling at an inflated price when members of the public group attempt to join the pump. Because of this characteristic, it is common for the peak trading volume to occur several minutes after the coin is announced on these private channels. So, although it is common for the coin to be announced on the full or half hour in these private channels, the time given for the trading volume to peak will vary between different pumps given how successful they are. A key design decision was made here to not filter the data points determined on the full and half hour time frame, but rather filter the data according to whether the coin had been mentioned in public telegram channels before or after being pumped.

3.5 Reddit scrape design

Several key design decisions were made when performing data collection on the Reddit social media platform. We focused on detecting discussions that involved coins on the Binance exchange, because all of our data points come from this exchange. The CoinMarketCap API was used to download a list of all BTC pairs listed on the Binance exchange in the form of their full name and symbol. For example, Bitcoin and BTC. This resulted in a list of 171 names and symbols [A.0.4 Table A.2]. Google's Big Query was used as a source of Reddit comment history. The Big Query platform has Reddit comment history publicly available, sorted by month in SQL tables. The supercomputer Theta at the Argonne National Laboratory in Chicago was used to run the data scraping scripts [A.0.5]. Due to processing power not being a restriction, we decided to initially scrape the entire comment history for coin names and symbols. Initial results showed that about 60% of all comments were being marked as a coin mention, which was far too many to be correct. It was discovered that this was caused by coin symbols such as "ONE" and "FUN" being normal words that are used in many different contexts. Upon removing these troublesome symbols, the amount of comments flagged reduced to 20%. This was still far too many comments, so it was decided to filter the comments based on subreddit. A subreddit is a subset of reddit dedicated to a particular community. The goal was to only flag comments which mentioned a coin name or symbol and came from a subreddit directly related to cryptocurrency trading.

In order to find a complete list of crypto based subreddits, Google's Big Query was used to find all subreddits were the words "coin", "crypto", "trade", "trading", "buy", "sell" and "currency" were mentioned in a comment in the month of January of 2019. The result was a list of 30,074 subreddits out of a total of about 1.2 million subreddits. This list still contained a very large number of non-crypto related subreddits, so further filtering was performed. The list was filtered by using the SQL query function "LIKE" which flags a string if another string is contained at a particular position in the query string. A initial list of strings to look for was constructed by including all of the subreddit names found on cryptolinks [2], which contains a list of the most popular cryptocurrency subreddits. From there, the list was extended to include the words of the exchange "Binance" and "crypto" so that any subreddit with the word "crypto" in its name would be flagged. This list would seem sufficient; however, the filter would still be missing the subreddit names that are directly related to the names or symbols of the coins. Therefore, the name of each coin was added to the SQL query so that subreddits that start with the name of the coin are flagged. Both the preserved case and lowercase version of the names were searched. Similarly, the symbol for each coin was added so that subreddits that start with the coin symbol in uppercase would be flagged. The result was a list of 1350 subreddits that matched the search criteria. In order to ensure that every listed subreddit was in fact crypto related, we manually visited each subreddit's webpage and deleted the ones that were invalid. After this filter we are left with 561 cryptocurrencies related subreddits that we used to filter comment data.

[2] `https://cryptolinks.com/reddit-cryptocurrency`

Chapter 4

Implementation

4.1 Peak detection algorithm

The peak detection algorithm used for collecting our data points was implemented in the python language using Jupyter notebooks to run the scripts in the browser. In order to gain access to the Binance API, I made a Binance account and acquired a pair of API access tokens to provide to the python-binance wrapper library [1]. The procedure for collecting pumped data points was implemented as follows: For each coin, we download the 30-minute interval candlestick data (open, high, low, close, volume) into the form of a pandas data frame. The trading volume column is then taken and run through the smoothed z-score peak detection algorithm. This peak detection algorithm returns a list the same length as our list of volume values containing the signal output for each timestep. The signal output is 0 if the volume was within 38 standard deviations of the moving mean, 1 if it was above 38 standard deviations and -1 if it was below 38 standard deviations. In our case, we are only interested in the points where the trading volume went from an output signal of 0 to an output signal of 1 which indicates a sudden significant increase in trading volume.

Figure 4.1 shows an example of the peak detection algorithm output. We then collect all of the timestamps where the signal went from 0 to 1. With this list of coins and timestamps of interest, we then download the candlestick data in the most precise format possible: 1-minute interval. Because we are interested in predicting these pump events, we download

[1] `https://github.com/sammchardy/python-binance`

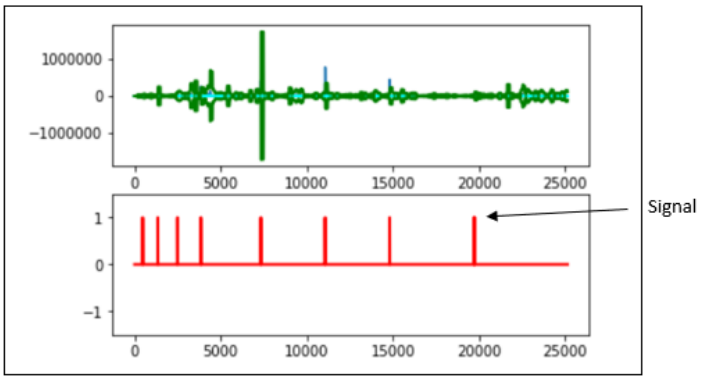


Figure 4.1: Peak detection output

the market data from 36 hours before the peak until 12 hours after the peak, which results in 48 hours of market data for each data point. The final step to collecting our data points is to download the aggregated order book data for each of the events. Finally, we output 2 CSV files named after the coin and the search date interval. 1 CSV for candlestick data and 1 for order book data. Data points that act as counter examples to pump events were also collected. These "non-pump" events, also known as normal market movement, were collected by running the peak detection algorithm for the same time span, but instead taking periods of 48 hour market data where the trading volume nether exceeded or dropped 38 standard deviations from the moving mean, meaning that the peak detection algorithm gave a constant 0 output.

4.2 Data verification

In order to verify our data points, a script was written that could pull out coin mentions from the historical chat history of the Telegram groups that were collected. Regex patterns were used to match common coin announcement message formats across different Telegram channels. In order to construct a list of common regex patterns, we manually scanned the messages for reoccurring coin announcement messages in a selection of the most popular Telegram channels in terms of number of members. 19 big Telegram channels were analyzed for coin message formats and a list of regex patterns was produced. These regex patterns can be seen in the appendix [A.0.3]. The historical chat history for each channel is stored in the form of a human readable html webpage. The python library BeautifulSoup [2] is a utility for scraping html files. This library was used to iterate through all of the message bodies in the html files. The script works by first extracting the date of the message from the title tag. The script then takes the message text and attempts to match all of the regex patterns [A.0.3] to it. If there is a match, then the next step is to extract the specific coin that is being mentioned. The format that the coin is written in each message varies widely. For example, a message may write the coin using a hashtag followed by all capitals, e.g. #BTC or they might just write the coin without the hashtag. Other coin formats include one capital letter followed by all lower-case letters, e.g. Btc or all lowercase letters. Because a coin can be any combination of letters ranging from 2 to 6 letters in length, we must first remove common words in messages that also match this format. Words such as 'Exchange', 'Buy', 'Target', 'This', 'Long', 'Short' and 'Open' are removed from the message text before scanning the message for the coin symbol. The most common way a coin is written is using all capital letters, so we first scan the message for words matching the regex "[A-Z]2,6". If no matches are found, we then attempt to scan using "[A-Z]1[a-z]1,5" which will match coins starting with a capital letter followed by all lowercase. If this fails, we finally attempt to find a word that matches "[a-z]2,6" which is all lowercase. Once the coin symbol is found, both the coin and the date are appended to a data frame. Upon inspection of the initial results, many of the coins and dates had duplicate entries. This can be explained by bots performing the coin announcements on multiple channels at the same time. Due to this, we chose to remove all duplicate entries before exporting the data to a CSV file. The pumped data points were validated by taking the timestamp of the highest trading volume for that data point and checking whether the coin was mentioned in the extract Telegram coin mentions within a given time frame. A non-pumped data point was validated by taking the middle timestamp (at the 24-hour mark) and checking whether that coin had not been mentioned on Telegram within a given timeframe.

[2] {https://pypi.org/project/beautifulsoup4/}

4.3 Binary classification

The next step was to build and test a variety of artificial classification models to determine whether pumped data points and non-pumped data points can be distinguished from each other. Due to the success of Logistic Regression and Random Forest models in [21], it was decided that these models would be evaluated. First, the data had to be processed into a form that could be feed into the artificial models. This was done by reading each instance into a data frame and then converting each instance into a 1-dimensional vector. This was done by appending each feature to a list and then appending the class label at the end. Each data point consists of 48 hours of data in 1-minute intervals which is 2881 values for each feature. Starting from 0 until 2881, each feature is appended to the vector. Finally, the class label is added; A pumped data point had a 1 at the end of each vector whereas a non-pumped data point had a 0. Next, the data was split into training and testing sets using sklearn's `train_test_split` method. A test set size of 0.33% of the total data was used and the sample was stratified on the class labels. Stratifying on the class labels results in the class weights becoming balanced to account for any class imbalances in the train and test split.

4.3.1 Binary classification at different data lengths

Next, we were interested to see just how many minutes of each data point was needed to get an accurate predication. In order to achieve this, we tested various different data lengths using the binary classification models and graphed the classifier accuracy. Because the peaks in the pumped data points are contained at the 36-hour mark, we decided to test data lengths ranging from 35 hours to 37 hours in 5-minute intervals. When converting the data points into vectors, only features within range of the length limits were added.

4.3.2 Prediction attempts with constructed features

Next, we attempted to construct some of our own features from the data points in an attempt to be able to perform a prediction of a class label using only data leading up to 1 hour before the peak in volume. Following the success of the results seen in [21], we have constructed features similar to the ones used in that paper. Table 4.1 on the following page describes the constructed features.

4.3.3 Adding features from order book data

In an attempt to achieve better performance, additional features were added from the order book data of each instance. First, the candlestick data and order book data for each data point was merged into a single CSV, while aggregating the order book data due to it being significantly larger in length. From the order book data, we could extract features such as the number of orders filled between each time step. We can also determine a Boolean value for whether the buyers placed the most orders rather than sellers and whether most of the filled orders were the best match in price for each time step. These additional features extend the features inspired by [21] by incorporating some features constructed from the order book data. A breakdown of these additional features are described in Table 4.2.

Feature	Description	X
Market Cap	The market cap of the coin retrieved from the CoinMarketCap API	-
Last Price	The price of the coin 1 hour before the pump event	-
Returns before pump	x-hour log return of the coin within the time window from x + 1 hours to 1 hour before the pump	1, 3, 12, 35
Volume before pump	Total amount of the coin traded within the time window from x + 1 hours to 1 hour before the pump	1, 3, 12, 35
Return volatilities before pump	Volatility in the hourly log return of the coin within the time window from x + 1 hours to 1 hour before the pump	1, 3, 12, 35
Volume volatilities before pump	Volatility in the hourly trading volume in coin within the time window from x + 1 hours to 1 hour before the pump	1, 3, 12, 35

Table 4.1: Constructed Features

Feature	Description	X
Number of orders	Total amount number of trade orders made within the time window from x + 1 hours to 1 hour before the pump	1, 3, 12, 35
Aggregate Buyer Makes Max	True / False for whether the buyers placed the most orders (rather than sellers) within the time window from x + 1 hours to 1 hour before the pump	1, 3, 12, 35
Aggregate Best Match Max	True / False for whether the most of the filled orders were the best match in price within the time window from x + 1 hours to 1 hour before the pump	1, 3, 12, 35

Table 4.2: Additional Order Book Data Features

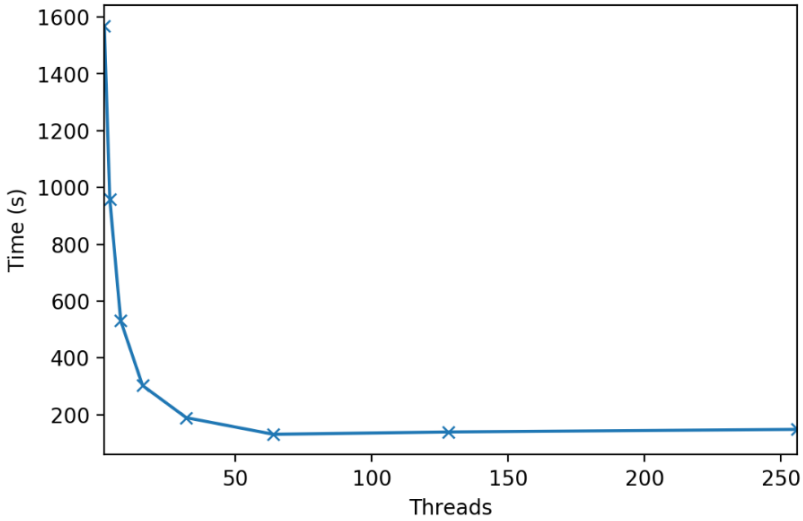


Figure 4.2: Multithreading results

Column name	Description
coin	The coin that was detected in the comment
subreddit	The subreddit the coin was posted in
date	The date in seconds that the comment was posted
comment	The content of the comment
names	An array of all coin names that were found in the comment
symbols	An array of all the coin symbols that were found in the comment

Table 4.3: Scraped Reddit data format

4.4 Reddit scrape implementation

The Reddit scraping script [A.0.5] was run on the supercomputer at Argonne National Laboratory near the University of Chicago. Each month of comment data in 2018 contains around 100 million comments, with months in 2019 coming closer to an average of 140 million. The Reddit script [A.0.5] was parallelized so that each month of data could be processed in parallel. We were able to process very efficiently using this technique. Figure 4.2 shows the time taken to process a subset of comments for 1 month versus how many threads are used to complete the computation. We can see that using up to about 60 threads makes the processing of each file much more efficient. The output for each month of data is written to a CSV file with the columns described in Table 4.3. We can then merge the resulting data frames for each month into one list of extracted comments.

4.5 Sentiment analysis implementation

The python library Vader Sentiment [3] was used in order to perform sentiment analysis on the extracted Reddit comments. Vader Sentiment is a lexicon and rule-based sentiment analysis tool that built specifically for sentiments expressed in social media. Vader is a good choice of too in our case because it does not require any training, it is already trained from generalized high stand lexicon. One of the biggest advantages of using Vader is that it is particularly good at recognizing emoticons and punctuation that indicate positive or negative sentiment. The use of emoticons and punctuation such as many exclamation marks is extremely common in social media discussion due to the informal nature of the chat platforms. Using the `SentimentIntensityAnalyzer` provided by this library, we can extract the polarity scores for a given comment. These polarity scores include a positive, neutral, negative a compound score. The positive, negative and neutral scores for a sentence all add up to 1 and give a percentage rating of the proportion of text in the comment that falls into each category. The compound score of a comment calculates the sum of all of the lexicon ratings that have been normalized between -1 (most negative) and 1 (most positive). The compound metric is therefore the most useful for determining whether a comment is either positive or negative by applying thresholds to the compound score. According to the documentation, a compound score of above 0.05 is a positive sentiment, a compound score between 0.05 and -0.05 is neutral and a compound score of below -0.05 is negative.

[3] `https://github.com/cjhutto/vaderSentiment`

Chapter 5

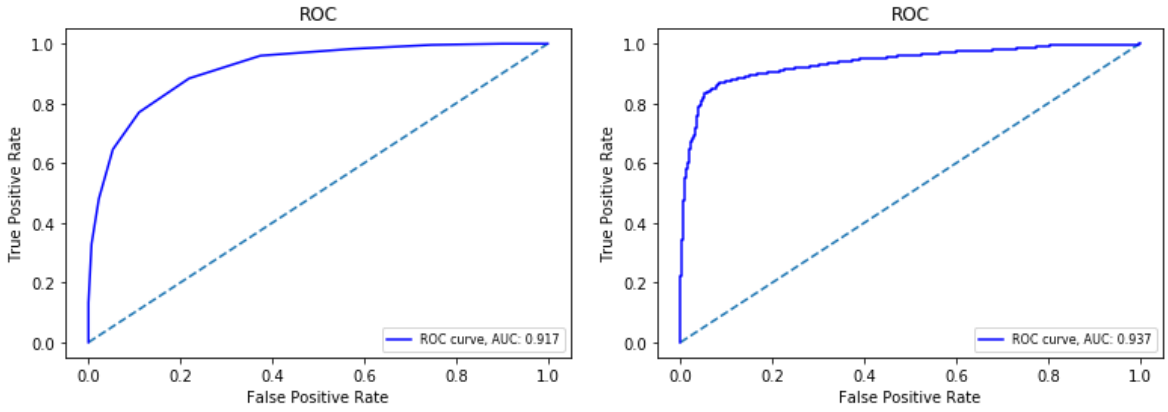
Evaluation

5.1 Data points collected

All data points were produced by running a smoothed z-score peak detection algorithm, as described in the implementation section, on Binance market history for 144 coins between 1st June 2017 and 1st June 2019. The result is 2086 pumped data points. Pump data was also received from the authors of the paper "The Anatomy of a Cryptocurrency Pump-and-Dump Scheme" [21] and when adding these pump events to our dataset, the result is 2196 pumped data points from coins and market history from the Binance exchange. A non-pumped data point is a 48-hour period of market history for a coin where there are no peaks in volume above 38 standard deviations from the moving mean. The result is 30,423 data points. These 2,196 pumped data points and 30,423 non-pumped data points is what we will refer to as the non-verified dataset. In order to create what we will call the validated dataset; the date and times of the pumped points are cross referenced with coin mentions on pump and dump Telegram group channels. A total of 7095 unique coin mentions from the 167 Telegram pump and dump channels were collected. A pumped data point was verified as being a valid pump if the time of the coin mention and the time of the peak in trading volume happened within 72 hours of each other. This resulted in 467 / 2196 of the pumped data points being verified. A non-pumped data point was verified in a similar way: If the coin was not mentioned in any Telegram channels within 72 hours of the middle of the market data, then it was considered normal market movement. This resulted in 24,179 / 30,423 of the non-pumped data points being verified.

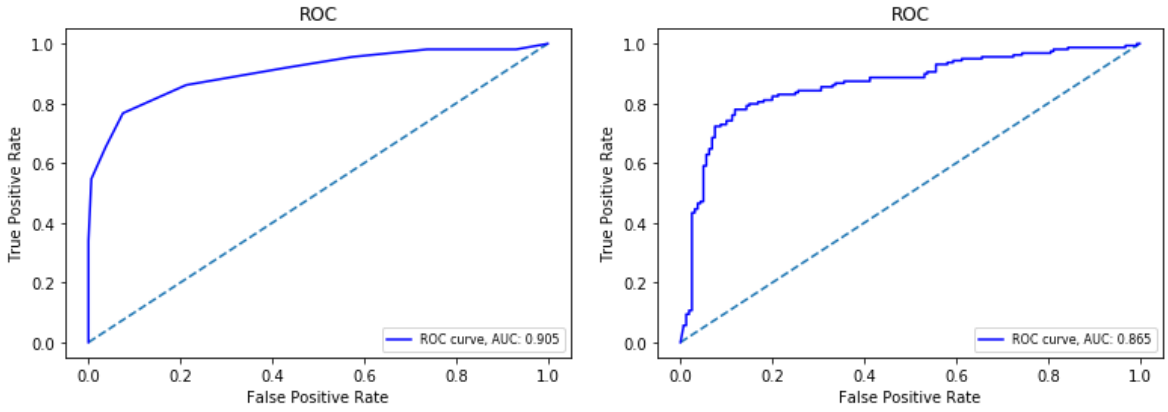
5.2 Binary classification results

In the first test, we attempt to simply classify pump event and non-pump event using Logistic Regression and Random Forest binary classification algorithms from the sklearn python library. First, we test the non-verified dataset. The dataset was split into training and test set using the sklearn `train_test_split` function. A test set size of 0.33% of the total data was used and the sample was stratified on the class labels. For the sake of time, a random sample was taken from the non-pumped data points so that the number of pumped and non-pumped instances were equal. The Random Forest classifier was able to achieve a test accuracy of 83.023% and a precision score of 0.87. The ROC curve for the Random Forest model is shown in Figure 5.1 (a). By analyzing the confusion matrix, we can see that the classifier correctly classified 577 / 749 of the pumped instances and 670 / 753 of the non-pumped instances in the test set. The Logistic Regression classifier was able to achieve a



(a) Random Forest Model (b) Logistic Regression Model

Figure 5.1: Non-verified dataset classification results



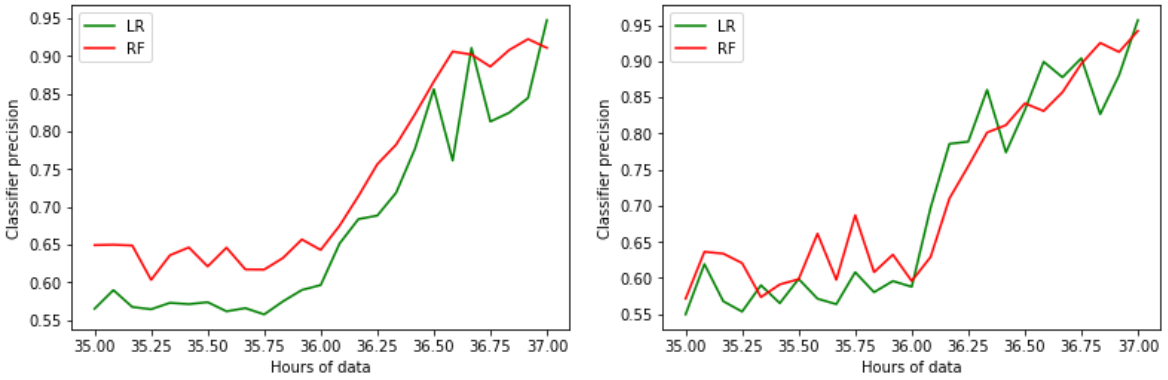
(a) Random Forest Model (b) Logistic Regression Model

Figure 5.2: Verified dataset classification results

test accuracy of 88.948% and a precision score of 0.92. The ROC curve for the Logistic Regression model is shown in Figure 5.1 (b). By analyzing the confusion matrix, we can see that the classifier correctly classified 636 / 749 of the pumped instances and 700 / 753 of the non-pumped instances in the test set.

Next, we test the same classifiers on the verified dataset. The Random Forest classifier was able to achieve a test accuracy of 84.639% and a precision score of 0.91. The ROC curve for the Random Forest model is shown in Figure 5.2 (a). By analyzing the confusion matrix, we can see that the classifier correctly classified 122 / 159 of the pumped instances and 148 / 160 of the non-pumped instances in the test set. The Logistic Regression classifier was able to achieve a test accuracy of 82.445% and a precision score of 0.87. The ROC curve for the Logistic Regression model is shown in Figure 5.2 (b). By analyzing the confusion matrix, we can see that the classifier correctly classified 122 / 159 of the pumped instances and 141 / 160 of the non-pumped instances in the test set.

The above results show that the classifiers are successful at differentiating the pumped and non-pumped events in both the non-verified and verified datasets with an accuracy above 80% in both cases. There seems to be no significant advantage when using the verified dataset over the non-verified dataset, however there may be a difference when attempting



(a) Non-verified dataset

(b) Verified dataset

Figure 5.3: Classification precision vs hours of data

to predict the events rather than simply classifying. We hypothesize this because pump and dumps organized by Telegram admins are more likely to have the coin selection based on a set of certain factors that make the selected coin "pumpable". These factors include things such as return volatilities and trading volume volatilities in the hours preceding the pump.

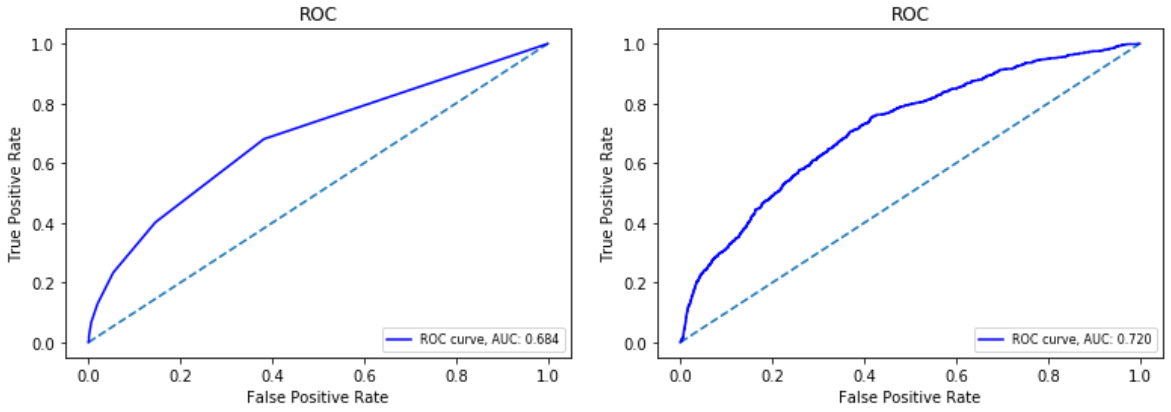
5.3 Classification results at different data lengths

Next, we experiment with reducing the amount of data we feed into the classifiers. In each data point, the peak is located at the 36-hour mark. Therefore, we test varying lengths of input data ranging from 35 hours to 37 hours to see how much of the data is needed for the classifiers to make an accurate classification. Figure 5.3 contains two graphs that show the relationship between the number of hours used for each instance of training / testing data point and the classifiers precision score for each dataset.

We can see that there is a sudden increase in precision at the 36-hour mark in both the non-verified and verified datasets, which is to be expected as this is where the peak in volume occurs. Unfortunately, the classifiers precision before the 36-hour mark is around 0.55 – 0.65 which indicates that the classifier cannot tell classify a pumped instance without seeing the peak in volume at the 36-hour mark. This means that we must try a different approach in order to attempt to predict the pump events before they happen.

5.4 Prediction results

Next we evaluate out prediction attempts using the features inspired by the paper [21]. These features include: Market Cap, Last Price and Returns before pump, Volume before pump, Return volatilities before pump and Volume volatilities before pump for various values of x hours before the pump. This results in a total of 18 features for each instance. First the non-verified dataset is tested. The Random Forest model achieved a precision score of 0.54. The ROC curve for the Random Forest model is shown in Figure 5.4 (a). By looking at the confusion matrix, we can see the model was able to correctly classify 21 / 724 of the pumped instances in the test set and 9656 / 9674 of the non-pumped instances in the test set. This means that a total of 703 / 724 of the pumped instances were incorrectly classified and only 18 / 9674 of the non-pumped instances were incorrectly classified. Next, the Logistic



(a) Random Forest Model

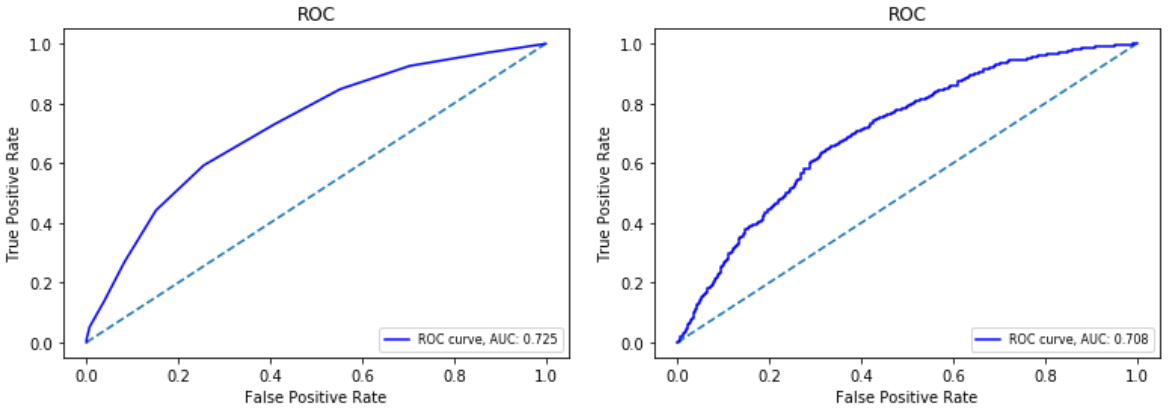
(b) Logistic Regression Model

Figure 5.4: Non-verified dataset prediction results

Regression model was tested. Again, the class weight parameter was set to "balanced". The Logistic Regression model achieved a precision score of 0.32. The ROC curve for the Logistic Regression model is shown in Figure 5.4 (b). By looking at the confusion matrix, we can see the model was able to correctly classify 94 / 724 of the pumped instances in the test set and 9474 / 9674 of the non-pumped instances in the test set. This means that a total of 630 / 724 of the pumped instances were incorrectly classified and 200 / 9674 of the non-pumped instances were incorrectly classified.

Next, the verified dataset is tested. The Random Forest model achieved a precision score of 1.0. By looking at the confusion matrix, we can see the model was able to correctly classify 2 / 154 of the pumped instances in the test set and all of the non-pumped instances in the test set. This means that a total of 152 / 154 of the pumped instances were incorrectly classified and none of the non-pumped instances were incorrectly classified. This is an obvious reduction in performance compared to the non-verified dataset because the classifier is classifying basically all of the data points as non-pumped. Similar results were observed using the Logistic regression model. The Logistic Regression model achieved a precision score of 0.04. By looking at the confusion matrix, we can see the model was able to correctly classify 12 / 154 of the pumped instances in the test set and 7673 / 7980 of the non-pumped instances in the test set. This means that a total of 142 / 154 of the pumped instances were incorrectly classified and 307 / 7980 of the non-pumped instances were incorrectly classified.

Due to the extremely poor results achieved by the base feature set, we added an extra set of features extracted from the order book [1.2] data of each instance. These features include Number of orders, Aggregate Buyer Makes Max and Aggregate Best Match Max at various values of x hours before the pump. Because order book data takes a long time to retrieve, the number of non-pumped data points was reduced to a random sample of 3623, which made the class labels much more balanced. The Random Forest model achieved a precision score of 0.64. The ROC curve for the Random Forest model is shown in Figure 5.5 (a). By looking at the confusion matrix, we can see the model was able to correctly classify 320 / 724 of the pumped instances in the test set and 1015 / 1196 of the non-pumped instances in the test set. This means that a total of 404 / 724 of the pumped instances were incorrectly classified and only 118 / 1196 of the non-pumped instances were incorrectly classified. The Logistic Regression model achieved a precision score of 0.58. The ROC curve for the Logistic Regression model is shown in Figure 5.5 (b). By looking at the confusion matrix, we can see the model was able to correctly classify 15 / 724 of the pumped instances in the test set and



(a) Random Forest Model

(b) Logistic Regression Model

Figure 5.5: Verified dataset prediction results

1185 / 1196 of the non-pumped instances in the test set. This means that a total of 709 / 724 of the pumped instances were incorrectly classified and 11 / 1196 of the non-pumped instances were incorrectly classified. As we can see from the prediction results, there is still far too many pumped instances being classified as non-pumped instances, which could be because of two reasons. One reason could be that our pumped dataset still contains some noise in the form of data points where trading volume increased but not from an organized pump, meaning that any factors that lead pump organizers to choose that coin are not present. This results in the classifier not being able to learn the difference between pumped and non-pumped instances because many of the features have similar values. The second reason could be that there is simply not enough difference in the features between the two classes, in which case we need to produce additional features that can separate the two classes more clearly. Results seen in paper [13] show that social media activity surrounding positive and negative coin discussion could help in predicting the pump events. In section 5.5 we explore results extracted from Reddit and in section 5.6 we discuss the potential to use sentiment analysis to construct additional features.

5.5 Reddit scrape results

Out of a potential 1.2 billion comments for the year of 2018, a total of 1,974,791 (1.9 million) were extracted using the Reddit scraping script [A.0.5]. That is a total of 0.165% of the comments for 2018. Reddit comment data for 2019 was only available up to and including the month of May. This is 5 months of data, with an average of 140 million comments per month which is a total of 700 million comments that were fed into the script for 2019. Of these comments, 340,500 were extracted which equates to 0.05% of comments extracted by the script.

When analyzing the comments picked up by our script, we would expect that coins with the highest market volume would be discussed the most on the social media platform. We expect this because coins with a high market volume are usually the most popular coins which means that more people are trading those coins on a regular basis. Table 5.1 shows the top 10 coins that were mentioned the most out of all of the filtered comments. Note that some coins have been indexed by both their full name and symbol, so for these coins both entries will be added together and considered as a single data point.

Coin	Comments
ETH + Ethereum	283,238
BCH + Bitcoin Cash	196,117
LINK (Chainlink)	154,096
XRP (Ripple)	117,083
NANO	114,303
EOS	87,059
SUB (Subtratum)	74,079
DATA (Streamr DATAcoin)	61,732
NEO	59,774
KEY	58,490

Table 5.1: Top 10 coins from Reddit data

Coin	Comments
ETH + Ethereum	283238
BCH + Bitcoin Cash	196117
LINK (Chainlink)	154096
XRP (Ripple)	117083
EOS	87059
NEO	59774
IOTA (Miota)	57708
XMR (Monero)	53814
VET (Vechain)	48326
XLM (Stellar)	38679

Table 5.2: Top 10 coins from Reddit with common words removed

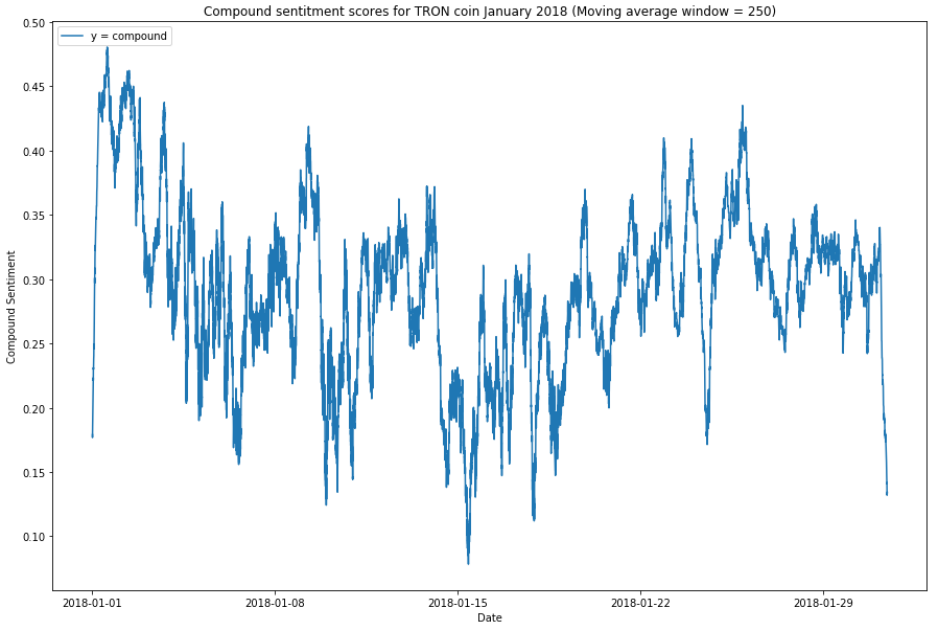
When looking at coin rankings in terms of direct volume on the CryptoCompare[1] website, we can see that 6/10 of the coins listed in Table 5.1 are included in the top 100 rankings. Direct volume is a measure of how liquid a coin is directly into the top list currency (Bitcoin). Therefore, coins with a high direct volume are being traded at a high rate. The data points not included in the top 100 rankings are: NANO, SUB, DATA and KEY. There is a clear similarity between these unranked data points, which is that the symbols are also normal words. This means that even though we are filtering the comments by crypto related subreddits, there is still some noise contained in the data produced by people both using these words normally and to refer to the coins they are discussing. In order to eliminate this noise, we remove these coins from the top 10 list and replace them with the next highest ranked coins. The resulting list is shown in Table 5.2. When comparing the new top 10 list of coins to the top 100 rankings on CryptoCompare, this time only one coin is not included: Vechain. The fact that 9/10 of the top 10 coins with the most amount of comments scraped are included on the top 100 direct volume list indicates that our script is effective at detecting and filtering comments discussing cryptocurrencies.

[1] `CryptoCompare.com`

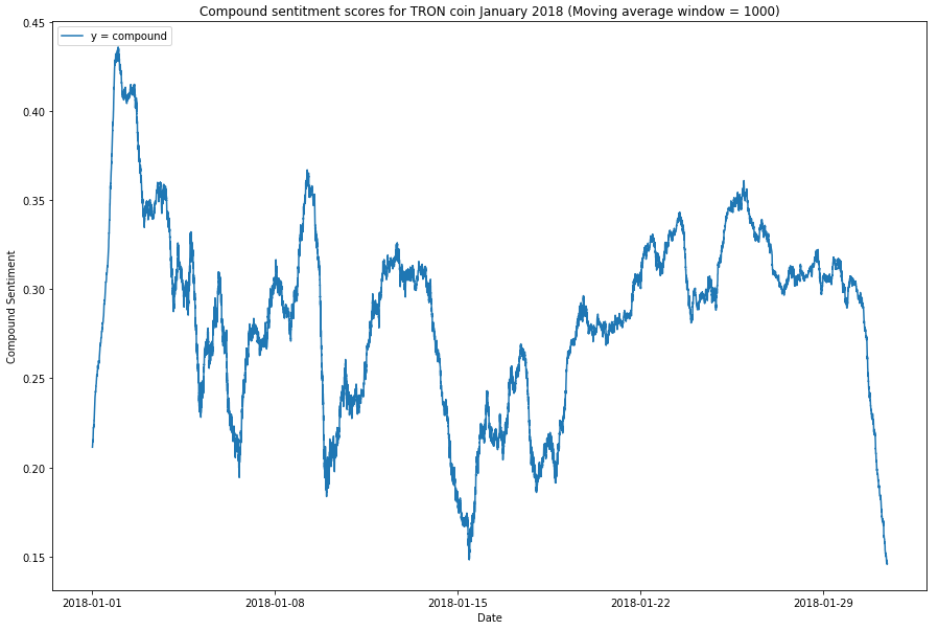
5.6 Initial sentiment analysis testing

Initial attempts have been made to perform sentiment analysis using the scraped Reddit comments. At this stage, only very basic testing has been performed by picking months of market data for coins with obvious indications of market manipulation and then extracting Reddit comments that were posted in this month and plotting the positive and negative sentiment values. One area of market data that caught our attention was the month of January 2018 for TRX (TRON) coin. During this month, a total of 33,396 comments were detected that contained the words "TRX" or "TRON" which is an average of 1077 comments per day. When referring back to our non-verified dataset of pumps, there are no data points that occur in this month. However, such a large amount of discussion surrounding the coin, in just the month of January, indicates that perhaps long-term interest in coins could be analyzed and utilized to predict more longer-term pump and dump events. Due to there being so many comments, it is necessary to graph the moving average of the compound sentiment score to be able to identify any patterns in positive or negative sentiment over time. Figure 5.6 (a) shows the moving average with a window size of 250 for the comments of January 2018 for the TRON coin.

The resulting plot is for a window size of 250 is still very sporadic. In order to make the trends in sentiment more easily visible, a moving window of 1000 was tested. As seen in Figure 5.6 (b), there is some clear potential to try and identify patterns and match these patterns to areas of manipulated market movement. There is potential to use the sentiment results to construct additional features for each pump and dump instance to try to achieve a higher accuracy in the attempts to predict these events. Previous research performed in "Social signals and algorithmic trading of Bitcoin" [7] by David Garcia and Frank Schweitzer outlines a good potential feature called 'Valance'. Valance refers to "the degree of pleasure or displeasure of an emotional experience" and in this case it is the positive or negative expression of a piece of text that mentions a specific crypto asset. In the field of psychological research, it is common to evaluate valance based on lexicon techniques [5], which is exactly how the python Vader library performs.



(a) Moving window = 250



(b) Moving window = 1000

Figure 5.6: Moving average of compound sentiment score at different window sizes

Chapter 6

Conclusions and Future Work

6.1 More Exchanges

The data points used in this research have been limited to the Binance exchange to keep the process of data collection as simple as possible. However, when observing pump analytics on PumpOlymp it is clear that pumps are being conducted on many more exchanges. For example, paper [21] indicated that a very large number of pumps were conducted on the Cryptopia exchange compared to other exchanges. From August to November 2018, 148 pumps were conducted on Cryptopia compared to 40 on Yobit, 24 on Binance and 8 on Bittrex. Cryptopia is the obvious next candidate for data collection, however in January 2019 the exchange was the target of a hack resulting in a major security breach. The outcome of the hack caused the exchange to lose a significant amount of the assets stored on the exchange. Reports indicate that up to 9.4% of Cryptopia's total holdings had been stolen, estimated to be about NZ $23 million dollars [10]. As of May 15, 2019, the company has gone into liquidation and it is unclear whether individuals who held currency on the exchange will get their coins back.

The next most intriguing exchange is Yobit. The current process of scraping Telegram coin announcement messages does not account for the exchange which the coin is being pumped on. This is due to regex patterns being difficult to match the exchange, because the format in which the exchange is mentioned across Telegram channels varies widely. We suspect that gathering pumped market data from the Yobit exchange will significantly increase the amount of data points that we can verify using our cross-referencing process. This is because of the large amount pumps identified on Yobit by Xu and Livshits in [21], where they found that 18% of their pumped data points came from Yobit while only 11% came from Binance. Unfortunately, the exchange with the most amount of data points in [21] was Cryptopia with 67%, which no longer exists. As of 20th September 2019, there are 569 altcoins paired with bitcoin on the Yobit exchange compared to 180 on Binance. There is a significantly larger number of altcoins on Yobit compared to Binance, which introduces more opportunities to manipulate coin prices. Incorporating data from various exchanges will also allow the artificial models training process to produce a more generalized model that can identify pump event from more than just the Binance exchange.

6.2 More social platforms

Future work for the project would also include scraping coin announcement messages from more social platforms other than Telegram and Reddit. A study performed in 2018 identified 3767 pump signals on Telegram and an additional 1051 on Discord between January and July for more than 300 different cryptocurrencies [8]. This shows that Discord could be a very useful data source for coin announcement messages and would enable us to verify a larger amount of data points. Work performed in the paper [13] indicates that Twitter is another social platform where pump and dump communities are advertised. Scraping Telegram links from Twitter posts could enable us to identify more Telegram channels and verify more of our data points.

6.3 Stricter data verification

When verifying the both the pumped and non-pumped data points, a threshold of 72 hours both before and after the peak in volume was used when comparing Telegram coin announcement messages. This is a very large threshold, especially because we expect that a coin will be announced very close to the time which it is pumped. Perhaps with more coin announcement data from a broader range of social platforms we could reduce this threshold to around 12 hours, as it is typical for the channel admins to post the profit results of the pump within the following day. We expect that this would greatly reduce the number of data points being verified because of the smaller threshold, however the data points that will get verified will be much better examples of pre-determined and organized events, which should allow the artificial models to pick up patterns in the market movement which motivate pump organizers to select that coin.

6.4 Real time data gathering

At this point in time, only historic market data ranging from 1st June 2017 to 1st June 2019 has been used to attempt predictions. Furthermore, historical Telegram channel data is severely limited due to channel admins deleting the old channels to hide any evidence that the pumps were conducted. This means that any data points that occurred before mid-2018 cannot be verified due to there being no available evidence in the Telegram data. This problem may be solved by using coin mentions across other social media platforms such as Reddit, Twitter and Discord but this is still unlikely due to pump organizers deleting the history. A better solution would be to design a script that can pull the historic market data for each chosen exchange at the end of each week. The same could be done for each of the identified social media channels. At the end of each week, the model could be re-trained including the new data points. With real time data gathering it will be possible to store historical data in large amount without having to worry about channel admins deleting the social data history. Having a large amount of data stored will produce an opportunity to analyze long term market movements corresponding to long term social sentiment.

6.5 Predicting long term pump events

This research has mostly focused on very short-term pump and dump events. However, with the results gathered using sentiment analysis on Reddit and Twitter data there could be an opportunity to identify pump and dump events that are carried out over a much longer time period. The peak detection algorithm can be altered slightly so that the moving window is 1 week rather than 12 hours and the threshold values can be altered to identify peculiar increases in trading volume over this longer time span. Perhaps there is opportunities to create features based on the sentiment analysis results that can be paired with market data features to train a model that can predict dump events of long-term pumping coins.

6.6 Conclusion

Our research backs up what many other research papers have found, that is pump and dump events can certainly be identified using both Logistic Regression and Random Forest models after they occur. Attempts to predict these events based on a set of constructed features relating to market cap, returns and volatilities in price and trading volume have appeared to be challenging, with no promising accuracy results being achieved. This research has proved that there is certainly manipulation that occurs in these cryptocurrency markets and all of the data needed to train predictive artificial models is publicly available, however it needs to be processed in such a way that produces data points that are obvious signs of organized manipulation. Initial attempts to gain additional features via the scraping and sentiment analysis of Reddit comments has proven that there is a large amount of discussion surrounding these coins occurring on social platforms. However, more research needs to be done in order to extract valuable patterns in the sentiment of these comments over time to produce features that that are indicative of potential pumps or dumps in the markets. We hope that the results presented in this research inspire more researchers in this field to extend this work by incorporating market data from additional exchanges and additional social media data from platforms such as Discord and Twitter. With a larger amount of both market and social media data, there is a huge potential to identify and expose both short-term and long-term pump and dump events in order to protect traders from losing investments due to fraudulent activity in the pump and dump community.

Bibliography

[1] ADVISORY, C. C. Beware virtual currency pumpand-dump schemes, 2018.

[2] BAUR, D. G., AND DIMPFL, T. Asymmetric volatility in cryptocurrencies. In *Economics Letters, 173:148–151*. 2018.

[3] CAO, Y., LI, Y., COLEMAN, S., BELATRECHE, A., AND MCGINNITY, T. M. Adaptive hidden markov model with anomaly states for price manipulation detection. In *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 26, NO. 2*. 2015.

[4] COINCENTRAL. Beginners guide: What is bitcoin?, 2019.

[5] D, G., AND F, S. Modeling online collective emotions. In *In Proc. of the 2012 Workshop on Data-driven User Behavioral Modelling and Mining from Social Media*. 2012.

[6] FRANKENFIELD, J. MS Windows NT kernel description, 2019.

[7] GARCIA, D., AND SCHWEITZE, F. Social signals and algorithmic trading of bitcoin. In *Royal Society open science*. 2015.

[8] HAMRICK, J., ROUHI, F., MUKHERJEE, A., FEDER, A., GANDAL, N., MOORE, T., AND VASEK, M. An examination of the cryptocurrency pump and dump ecosystem. *SSRN* (2019).

[9] JAHANI, E., KRAFFT, P. M., SUHARA, Y., MORO, E., AND PENTLAND, A. Scamcoins, s*** posters, and the search for the next bitcoin^{TM}: Collective sensemaking in cryptocurrency discussions. In *Proceedings of the ACM on HumanComputer Interaction, 2:79*. 2018.

[10] JUNN, J. How new zealand company cryptopia lost over $20 million from a hack', 2019.

[11] LEANGARUN, T., TANGAMCHIT, P., AND THAJCHAYAPONG, S. Stock price manipulation detection based on mathematical models. In *International Journal of Trade, Economics and Finance, vol. 7, no. 3, pp. 81-88*. 2016.

[12] MARTINEAU, P. Inside the group chats where people pump and dump cryptocurrency, 2018.

[13] MIRTAHERI, M., HAIJA, S. A.-E., MORSTATTER, F., STEEG, G. V., AND GALSTYAN, A. Identifying and analyzing cryptocurrency manipulations in social media. In *arXiv:1902.03110v1 [cs.SI]*. 2019.

[14] NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system.

[15] RAJESH, A. K., AND WU, G. J. Stock market manipulation-theory and evidence. In *University of Michigan Business School Working Paper*. 2004.

[16] RAMOS, C., AND GOLUB, N. Cryptocurrency pumping predictions: A novel approach to identifying pump and dump schemes. In *http://cs229.stanford.edu/proj2017/final-reports/5231579.pdf*. 2017.

[17] SETH, S. Market capitalization defined, 2019.

[18] TAO LI, DONGHWA SHIN, B. W. Cryptocurrency pump-and-dump schemes. *SSRN* (2019).

[19] TUWINER, J. 9 best bitcoin & cryptocurrency exchange reviews (2019 updated), 2019.

[20] WILLIAMS-GRUT, O. Meet the crypto trader who says he bought a tesla with 'pump and dump' profits but claims the scams aren't bad: 'it's a game', 2017.

[21] XU, J., AND LIVSHITS, B. The anatomy of a cryptocurrency pump-and-dump scheme. In *arXiv preprint arXiv:1811.10109*. 2018.

Appendices

Appendix A

Appendix

A.0.1 Project Proposal

Introduction

There is a big concern in the crypto community that some markets are turning "rotten", meaning that market prices are being manipulated by groups of people in order to turn a quick profit. Schemes called "pump and dumps" are often carried out on penny stocks (crypto coins under one dollar) in which the price of the coin is pumped up by manipulators and later dumped when they sell all of their shares when the price is high. This project will investigate the pattern of pump and dump schemes and attempt to build more predictive tools that can identify pump and dump events before the effected market is dumped.

The Problem

There is an ever-growing concern that Bitcoin and other crypto currencies are frequently manipulated by groups looking to make a quick profit. Small time investors are put at risk from buying in to false positive statements on social media, only to buy shares at a high price and lose their investment when the market is dumped by the market manipulators. This project will attempt to demonstrate a relationship between social media activity and crypto prices, so that potential pump and dump events can be identified and avoided.

Proposed Solution

The project will attempt to identify a common pattern that pump and dump events look like by analyzing crypto price data with python scripts. Visualizations of the events in the form of graphs will be developed and analyzed for patterns. (est: 2 weeks)

Social media posts and comments on forums such as Reddit will be downloaded and analyzed with text classification scripts, for indications of groups attempting to promote certain crypto coins. (est: 2 weeks)

The project will then attempt to match social media influence with pump and dump events in order to find a relationship between the timing of the social media posts and the dumping event of the coin. (est: 3 weeks)

The final product will be a python script which can analyze real time market price data for a chosen coin, and use social media hints to give a percentage chance that a dump will happen within a certain time frame. (est: 3 weeks)

Evaluating your Solution

The program will be evaluated by running the scripts on sets of historical data that are confirmed to be pump and dump events. If the script can predict pump and dump events with a high enough certainty, then it will be considered successful. The script will also be tested on non-pump and dump data to confirm that it does not produce false positives.

Resource Requirements

The only resources required for the project will be a personal computer with Python and Jupiter Notebook installed. Python libraries for collecting social media data and crypto price data will also be used.

A.0.2 Telegram Channels

Bulls Eye Signals	BIGGEST CRYPTO INVESTMENT	Crypto Trader™
Cryptocurrency: Wolf Posting	ELITE PUMP GLOBAL CHANNEL	Binance Signals
Altcoin Pumps	Altcoins Booster Community	Crypto Bulls Pump ®
A+ Signals – Bitmex & Binance	WEB Pump YoBit	Crypto Pump Up
McAfee Alt Signals™	COINLANCER PUMPER	Arabic Big Pump
WORLD CRYPTO COMMUNITY	Binance Pump Signal®	Crypto Rocket ®
GALAXY PUMP	Crypto Hot Signals™	Crypto of the Day
MoneyPumps VIP	Binance Announcements	Crypto VIP Paid Signal™
DA PUMP	Banana Pump	Binance Mega Pump
Creative Signals	BWP(baby whale pump)	BEE Signal ™
Cryptology	Top Pump	Extreme_Pumps_CE
Cryptopia Pump	Bull Signals	Binancian Signals
Crypto Bulls Pump	PIRATES PUMPS	Franklin Pump.
Partners Of Cryptos	CryptoCoinRankings Private Signals	Binance Signals ®
Cryptonians	Binance Profit Signal	Binance Daily Signals!
Crypto God	CryptoHunter |	Crypto Life Margin
Crypto Profits	TOP PUMP VIP®™	Crypto Trading.
Crypto Future Signs	Bullish Signals	Crypto VIP signal™
Call Of Pumps	Super pumps	Crypto Pump
DUTCH CRYPTO PUMPS!	CAMP	American whales us
Crypto Coins	DAILY TRADE SIGNALS	Dragon Signals
Crypto God's	Dr. Crypto - Pumps & Signals	Coin|Coach|Signals
Central Pumps	©Pumpin®Time™	premium binance signals turkey
Crypto pumpers	Big Pump Signals	Fairwin Crypto News/Pump Signals
Crypto Advisor	Crypto signal channel	Daily Crypto Profits
All Link new bot	Crypto Signals | Smart Investments	Fast Crypto Signals.
Crypto Profit Coach™	Mega Pump Group	Caesar's Scalpers
Crypto Signals Official	COINEXCHANGE Whales Trade Group	Crypto Coinsultants
Hot Signals Binance Bittrex	SPARTA PUMP TEAM	Crown Signal Notifier
Crypto Free Signals	Elite Crypto Group	FAT BULL CRYPTO FOREX
Crypto God Signal™	OSNOVA PUMP	Wealthy Whale Pumps
PumpMyWallet	Bitcoin Pump Group	®Crypto Guru Bittrex Signals
Big signal	Exposure Pumps	Wolfsignalpump
Pump Club - Yobit	CryptoFamily New	Big Crypto Pump
Crypto Family Pumps	Crypto Expert.	Pump Masters
PumpWhales	Crypto Analysis Official™	Cryptopia Family Pumps
Dragon's Lair Signals	Binance and Cryptopia Pumps	Baby Whale
Eternal Crypto Pumps	Bomba bitcoin c̈ryptopia¨	Crypto Toros
GAINS Private Group (G.P.G)	PUMP MASTERS	Crypto Pumps
Best pump group	Big Pump channel	Binance And Cryptopia Pumps
Binance Pump Whales	Bitcoin Pump VIP	Crypto experts signal
Yobit Pump Team	Crypto Pump Signals	Genuine_Callz
The pumping army	Big Pump Signal	Pump Up
Free-For-All Pumps	Donald Pump	Great Big Pumps
BULLS PUMP	Cryptoverse	European Pumps
CRYPTO PUMP	Golden Ticket Pumps	Bigpump24
Crypto Elite Signals	Crypto Pump Island	Crypto Trading Expert
2 PUMPS EVERY DAY	Ultra Pumps™	Crypto Insiders
Crypto Watch	Crypto Pump Squad	Crypto Warrior
Dragon Pumps & Signals	CRYPTO COINS TRADING®	Crypto Market Signalz
20X Dicebot -scripts	EAGLE PUMPS	PumpingHard
BigPumpGroup.com	Crypto Signals & Pumps	CRYPTO BILLIONAIRE
Palm Venice Beach	Global Pump Signals	Bittrex - BigPumpGroup.com
Goat Pumps	Crypto_Pump	Pump Signals
F14sH_Pumps		

Table A.1: Telegram Channels used for coin announcement scraping

A.0.3 Coin announcment regex patterns

- "Buy Coin [U00010000-U0010ffff]+ [U00010000-U0010ffff]+ #[A-Z]2,6[U00010000-U0010ffff]+"
- "Coin name : [A-Z]2,6"
- "COIN IS : [A-Z]2,6"
- "Coin: [A-Z]2,6"
- "coin is : [A-Z]2,5"
- "COIN: #[A-Z]2,6"
- "COIN NAME - [A-Z]2,6"
- "The coin we picked is #[A-Z]2,6"
- "Buy #[A-Z]2,6 (Binance)"
- "Buy #[A-Z]2,6 under"
- "Buy #[A-Z]1[a-z]1,5 under"
- "Buy [A-Z]1[a-z]1,5 under"
- "Buy [A-Z]1[a-z]1,5 current"
- "Buy [A-Z]2,6 current"
- "Buy #[A-Z]2,6 at"
- "Buy #[A-Z]1[a-z]1,5 at"
- "Buy #[A-Z]2,6"
- "Buy #[A-Z]1[a-z]1,5"
- "#[A-Z]2,6 Buy zone"
- "#[A-Z]2,6 BUY"
- "#[A-Z]1[a-z]1,5 Buy zone"
- "#[A-Z]2,6 *Binance*"
- "#[A-Z]1[a-z]1,5 *Binance*"
- "#[A-Z]2,6 $BINANCE$"
- "#[A-Z]1[a-z]1,5 $BINANCE$"
- "#[A-Z]2,6 on #BINANCE"
- "#[A-Z]2,6 on #Binance"
- "#[A-Z]2,6 short"
- "#[A-Z]2,6 Short"
- "$[A-Z]2,6"

Litecoin	Basic Attention Token	SALT	Zilliqa
XRP	Horizen	Cardano	Polymath
Dogecoin	Aeternity	Viberate	Bluzelle
Dash	IOTA	Everex	WePower
Groestlcoin	SONM	Cindicator	Ren
Monero	Bancor	Enigma	Nucleus Vision
CloakCoin	KingN Coin	Eidoo	POA Network
Bytecoin	FunFair	AirSwap	TrueUSD
NavCoin	Status	BlockMason Credit Protocol	Ontology
BitShares	EOS	Aion	Ravencoin
Viacoin	AdEx	Request	Loom Network
Stellar	Storj	Ambrosus	Pundi X
Syscoin	Crypto.com	Bitcoin Gold	Wanchain
Verge	Gas	NULS	Mithril
Nexus	Metal	Ripio Credit Network	Dock
NEM	Populous	ICON	KEY
Ethereum	OmiseGO	Red Pulse Phoenix	IoTeX
Siacoin	Civic	Etherparty	QuarkChain
Augur	Ethos	Enjin Coin	GoChain
Decred	Bitcoin Cash	Power Ledger	Mainframe
PIVX	Binance Coin	Streamr DATAcoin	VeChain
Lisk	OAX	Aeron	Blocktrade Token
DigixDAO	district0x	Genesis Vision	WinToken
Steem	Blox	Quantstamp	Paxos Standard Token
Waves	Dent	Bitcoin Diamond	USD Coin
Comet	0x	Time New Bank	Fantom
Ardor	YOYOW	Aave	Bitcoin SV
Ethereum Classic	HyperCash	CyberMiles	Menlo One
Stratis	Nebulas	Tael	BitTorrent
NEO	Tierion	Gifto	Fetch
SingularDTV	Waltonchain	OST	Ankr
Zcoin	Loopring	Storm	Cosmos
Atomic Coin	Po.et	aelf	Celer Network
Zcash	Monetha	Bread	Theta Fuel
Golem	Agrello	QLC Chain	Scopuly
Wings	Moeda Loyalty Points	AppCoins	CONUN
Komodo	Neblio	Insolar	Matic Network
Nano	TRON	Selfkey	Harmony
Ark	Decentraland	IOST	Bitcoin BEP2
Skycoin	Chainlink	THETA	Algorand
iExec RLC	Kyber Network	SingularityNET	Elrond
Lunyr	VIBE	ChatCoin	Dusk Network
Qtum	Substratum		WINk

Table A.2: Coins from Binance searched for in the Reddit scraping script

A.0.4 Coins from Binance searched for in Reddit script

A.0.5 Reddit scraping script

```
if len(sys.argv) != 3:
        print('invalid args')
        print('please use 2 agrs: "month-year" and "number of files"')
        exit()

date = sys.argv[1]
number = sys.argv[2]

names = []
with open('binance_names.csv', mode='r', encoding='utf-8-sig') as csv_file:
        csv_reader = csv.reader(csv_file)
        for row in csv_reader:
                names.append(row[0])
symbols = []
with open('binance_symbols.csv', mode='r', encoding='utf-8-sig') as csv_file:
        csv_reader = csv.reader(csv_file)
        for row in csv_reader:
                symbols.append(row[0])

coin_mentions = pd.DataFrame(columns =['coin', 'subreddit', 'date', 'comment'])
c = [], b = [], s = [], d = []

for n in number:
        num = str(number)
        df = pd.read_csv('gs://reddit-comments-' + date + '/comments000000000' + num,
                    compression='gzip')
        for index, row in df.iterrows():
                for name in names:
                        if isinstance(name, str) and isinstance(row['body'], str):
                                if " "+name.lower()+" " in row['body'].lower():
                                        b.append(row['body'])
                                        c.append(name)
                                        s.append(row['subreddit'])
                                        d.append(row['created_utc'])
                for symbol in symbols:
                        if isinstance(symbol, str) and isinstance(row['body'], str):
                                if " "+symbol.lower()+" " in row['body'].lower():
                                        b.append(row['body'])
                                        c.append(symbol)
                                        s.append(row['subreddit'])
                                        d.append(row['created_utc'])
coin_mentions['coin'] = c
coin_mentions['subreddit'] = s
coin_mentions['date'] = d
coin_mentions['comment'] = b
coin_mentions.to_csv(date + ".csv")
```