Opinion Mining and Social Media Sentiment Analysis in the Prediction of Cryptocurrency Prices

School of Mathematical, Physical and Computational Sciences

Indivdual Project - CS3IP16

Student: Andrew Sotheran
Student Number: fr005432
Supervisor: Kenneth Boness
Word Count: Place Holder

Submission date: Place Holder

Abstract

The volatility of the stock markets is an aspect that is both hard to predict and to mitigate particularly when relating to the cryptocurrency market. Commodities such as cryptocurrencies are profoundly volatile and have attracted investors in an attempt to make quick profits on the market. These financial commodities are subject to the whim of public confidence and platforms such as Twitter and Facebook are most notably utilised to express opinions. Extrapolating sentiment from such platforms has been used to gain insight into topics across industries, thus applying it to crypto-market analysis could serve to show a relationship between public opinion and market change.

This project looks into public perception of the cryptomarket, by analysing Bitcoin-related tweets per hour for sentiment changes that could indicate a correlation to market fluctuations in the near future. This is achieved by training a recurrent neural network on the severity changes of historical sentiment and price over the past year every hour. The predictions are then shifted forward in time by 1 hour to indicate the corresponding Bitcoin price interval.

Acknowledgements

I would like to express my gratitude to Dr. Kenneth Boness for his continued support and guidance throughout this project.

Secondly, I want to express gratitude to PhD. Jason Brownlee, of Machine Learning Mastery for having clear and thorough explanations of machine learning concepts and metrics.

I would also like to thank my family for their support during the development of this project.

Glossary

Bull(ish)/Bear(ish) Markets - Relates to a trend of the market price increasing and decreasing respectively

Highs/Lows - The highest and lowest trading price of a giving period

Fiat Currency - A currency without intrinsic value that has been established as money

BTC - Bitcoin's stock symbol

Twitter - Online social media platform, which allows users to post information or express opinions through messages called "Tweets"

Tweets - The name given for messages posted on the Twitter platform, which are restricted to 280 characters.

Hashtag - Is a keyword or phrase used to describe a topic and allows the tweets to be categorised.

Fomo (Fear of Missing Out) - Is used to describe buying behaviour when stocks are moving suddenly and more buyers appear to enter all of a sudden.

Shorting - Or short sale, is the sale of an asset that the investor buys shares and immediately sells them, hoping to make a profit from buying later at a lower price.

Doubling Down - Is to take further risk on a stock by doubling effort/investment in a hope and attempt to raise the price

RNN - Recurrent Neural Network

LSTM - Long-Short Term Memory Neural Network

Contents

Introduction

The premise of this project is to investigate into whether the sentiment in social media has a correlation to the prices of cryptocurrencies and how this could be used to predict future changes in the price.

The chosen cryptocurrency that will be focused in this project will be the currency that has the most community and backing and has been known to lead other fiat currencies, Bitcoin (BTC). Bitcoin is seen as one, if not the first cryptocurrency to bring a wider following to the peer-to-peer token transaction scene since 2009. Although it was not the first token to utilise blockchain technology, it allowed investors to openly trade a public cryptocurrency which provided pseudonymous means of transferring funds through the internet. Thus it has been around longer than most of the other fiat currencies and is the most popular crypto-token due to it's larger community base.

Most financial commodities are subject to the whim of public confidence and are the core of it's base value. A platform that is frequently used for the public to convey their opinions on a commodity is that of Twitter which provides arguably biased information and opinions. Whether the opinions present a basis in facts or not, they are usually taken at face value and can influence the public opinion of given topics. As Bitcoin has been around since 2009 the opinions and information on the commodity are prevalent through the platform. In the paper *Sentiment Analysis of Twitter Data for Predicting Stock Market Movements* by *Majhi et al.* [1] 2.5 million tweets on Microsoft were extracted from Twitter, sentiment analysis and logistical regression performed on the data yielded 69.01% accuracy for a 3-day period on the increase/decrease in stock price. These results showed a "*good correlation between stock market movements and the sentiments of public expressed in Twitter*".

The background of this project is in response to the volatility of the cryptocurrency market, which can fluctuate at a moments notice and can be seen to be social media driven. The history of the price of Bitcoin and what was being discussed on the currency around it's most volatile period to-date, Nov-2017 to Feb-2018, shows a strong bullish trend which saw Bitcoin reach a $19,500 high in mid-Dec. While social media, such as Twitter, during that period was had an extremely positive outlook on the cryptocurrency. The trend was short lived and saw the market crash only a month later, with only a couple of sell-offs, expected for the holidays rush, accompanied by negative outlooks posted on social media turned the market against itself which saw the longest bearish market in Bitcoin's history and is still trying to recover today.

Due to how volatile the crypto-market can be, there is a need to either mitigate or to anticipate where the markets are heading. As the crypto-market and Bitcoin are affected by socially constructed opinions, either through Twitter, news articles or other forms of media, there is a way to perform the latter, where the prices of Bitcoin could be predicted based on the sentiment gathered from social media outlets.

The aim of this project is to create a tool that gathers tweets from Twitter, obtains the overall sentiment score of the given text while gathering historical price data for the time period gathering occurs. Features are then extracted from the gathered data and used in a neural network to ascertain whether the price of the currency can be predicted from the correlation between the sentiment and price history of the data.

This report will discuss the justifications for the project and the problems it will be attempting to resolve, the stakeholders that would benefit the most from this system and what this project will not attempt to accomplish. Similar tools will be critiqued and examined for their feature set and credibility in the literature review along with current sentiment analysers, algorithms, natural language processing techniques and neural networks in their respective topics and comparing their accuracy for this project purpose. The solution approach will discuss the decisions and reasoning behind choosing the techniques and tools used for this project and will outline the requirements for this project. Implementation of the chosen techniques and tools, with the discussion of important

functions of the system will formulate the implementation section of this report with an in-detail explanation of the function's use and data flow of the system.

Problem Articulation

Problem Statement

The key problems this project attempts to address are that of, an open-source system available to the public that aids in the analysis and prediction of BTC. The accuracy of open-source tools and technology when applied to the trading market scene and to identify whether there is a correlation between Twitter sentiment and BTC price fluctuation. While there are existing tools only a few are available to the public and only provide basic functionality, while others are kept in-house of major corporations who invest into this problem domain.

The other issue presented here is that assuming perfect accuracy can be achieved is naive. As this project will only be using existing tools and technologies thus, there are limitations to accuracy that can be obtained. One of that being the suitability of the tools, there are no open-source sentiment analysers for stock market prediction, thus finding a specifically trained analyser for the chosen domain in highly unlikely. In relation, finding the most suitable machine learning or neural network is equally important as this will determine the accuracy of the predictions. Due to being a regression problem, machine learning techniques and neural networks that focus around this and forecasting should be considered.

The accuracy and suitability of various machine learning methods and neural networks are a known issue in their respective domains, this investigation should be carried out to determine their suitability for their needed use in this project and should be detailed in the literature review.

This project will focus on the investigation of these technologies and tools to justify whether it is feasible to predict the price of BTC based on historical price and the sentiment gathered from Twitter. Limitations of the system and it's accuracy in predictions should be investigated and discussed to determine the implemented solution is the more suitable compared to other methods.

Stakeholders

The main stakeholders of this system would be those looking to invest in the cryptocurrency markets, in this projects regard, specifically into Bitcoin.

- Public Investors - These are investors from the general public. These investors can decide to either actively or passively invest in the markets but are essential for the general use of a given cryptocurrency. This type of investor would benefit the most from an open-source system such as this, as it will aim to provide a basis for decisions for buying or selling Bitcoin. Additionally, due to the lack of any open-source tools available, these stakeholders could be seen as being left in the dark when it comes to predicting the direction of Bitcoin where Businesses and Enterprises will have a one up, due to having an internal system for predictions.
- Speculators - These stakeholders can be both public and business, who aim for the chance of the possibility fast. They actively invest at points where a market is an impending rise in price and tend to sell after a market makes them a reasonable amount of money before it possibly drops. These stakeholders would benefit from such a system as it will provide a means to identify and predict short term gains in the Bitcoin market, and if taken into decisions could make a profit.
- Business Investors: These will be investors of a business who would be investing on the behalf of a company. A system such as that this project will provide may benefit such a stakeholder, but the information would be used as a collective with others to justify the investment. Additionally, this system may not benefit this stakeholder as the company they are investing for may have an equivalent or better system.

- Prospect Investors: These are new investors to the cryptomarket scene who are looking to get into the market and are generally looking for initial information of the market movement. This system will benefit such a stakeholder in their initial decisions in market investment, but not as much as a generally more active investor. This is due to the extent to which a new investor invests compared to a establish active investor.
- Developer - Andrew Sotheran: The developer responsible for this project by developing a solution that satisfies the problem and objective defined in the *Technical Specification*. As the sole developer of this project it should be ensured that the system is developed on time and the project runs smoothly.
- Project Supervisor - Kenneth Boness: Is the projects supervisor whom will oversee the development through weekly project meetings. Weekly feedback will be given on the progress and direction of development, and will offer advice to ensure the quality of the solution.

Project Motivation

The motivation behind the project stems from a range of points, from personal and public issues with the volatility if the crypto-market, and how losses specifically could be mitigated. The personal motivations behind the conceptualisation of this began two years ago during the crash of late 2017-2018, which saw new investors blindly jump into the trend that was buying cryptocurrencies. During this period of November to December 2017 saw Bitcoin's price reach $20,000 from $5,000, new public investors jumped on the chance to buy into the trend of possibly making quick profits and the fear of missing out (FOMO). In late December, a few holiday sell-offs occurred from business and big investors, this coupled with a few negative outlooks posted on social media by news outlets caused the market to implode causing investors to panic sell one after the other and posting negativity on social, thus causing more decline in the market. As a result, this caused personal monetary loss and distress as being a long-term investor.

Another motivation is that at the time of writing, there are no pubically available systems that combine sentiment analysis with historical price to forecast the price of Bitcoin or any other cryptocurrency. There are papers and a few code repositories that implement a similar concepts [2] - *Use of a Multilayer Perceptron network for moving averages in Bitcoin price*, [3] - *Predicting Bitcoin price fluctuation with Twitter sentiment analysis*, [4] - *Predict Tomorrows Bitcoin (BTC) Price with Recurrent Neural Networks* but are not operational. System such as [1] hosted on Coingecko, a popular cryptocurrency track site, provides a tool for basic sentiment analysis but doesn't give an evaluated indication of the direction of the market as a prediction. This leaves the public to the whim of volatility of the market without a means to know what the next, say an hour, could entail to possibly reduce losses if the market drops. Such system are usually kept in-house of major corporations whom invest significant time into tackling such a problem. Additionly, this could be seen as a positive for major investors, as such a system could cause panic selling if public investors soley trusted such a system.

Technical Specification

This project will need to follow a specification to ensure that the quality and the problem statement is met. This section will outline what this project should include, what it will not consist of and will guide the development of this project.

General:

- To investigate into the use of lexicon-dictionary based sentiment analyser approach in for sentiment analysis and it's customisability for a given topic domain
- To create a system that can predict the next hour of Bitcoins price when given the price and sentiment for the past hour
- To investigate into natural language data pre-processing techniques and how these could be used to filter out unwanted data
- To investigate into the use of a neural network, specifically an LSTM for forecasting price data
- Ultimatly, to investigate into how the use of sentiment effects the prediction of price for the next hour

Natural Language pre-processing (Spam and language detection filtering)

- To produce a system that processes the historical and live tweets, removing unwanted characters, removing urls and punctuation.
- To produce a system for spam filter using probability likelihood for processed tweets. A naive Bayes approach may be suitable for this given task
- To produce a language detection and filtering system that removes all tweets that are not of the English language or containing non-basic-latin characters
- To provide a means for stemming, tokenisation and stopword removal to aid in data pre-processing for language detection and spam filtering

Neural Network

- To produce a neural network which trains on collected, historical and live data, to forecast the future price of Bitcoin, based on price and sentiement
- To produce a neural netowork which accomplished the same as the other above, but with out use of sentiment
- To produce metrics to justify accuracy of the model
- To produce data files containing, the current time of predictions alongside current hour price and sentiment. This should also include a suggested action based on a threshold for the price difference between hours.
- To produce JSON files containing the true and predicted price values of every hour for trained data, and another for current reoccuring predictions.

Interface

- To produce a basic interface which displays the predicted values alongside true price values with a time interval step of an hour. This can be displayed as both a table consisting of:

  Date of prediction, predicted price of next hour, current hour price and sentiment, and a suggested action based on a threshold for the price difference between hours.

  To produce charts displaying the true and predicted price values for every hour, from both start of new predictions made, and from training predictions

- To display a table of performance metrics of the trained model

Server

- This system, both prediction system and interface, should be deployed to a server due to the need to be constantly running

Project Constraints

This project will not attempt to justify the accuracy of the chosen algorithm or tools over other algorithms. It will be discussed in the solution approach the justifications made on why the chosen algorithm and tools have been used for this project over the others, but accuracy will not be directly compared.

This project will only be coded to predict an hour ahead as the model will be trained on an hourly basis as the data is gathered per hour. Predictions for further in the future can be modelled but will be seen as a future improvement to the system.

The detail of a interface may be subject of change through this project due to time contraints and the focus being the investigation of the impact social media has on market predictions.

Literature Review

Existing Tools

An aspect that this project will be attempting to address is that, at the time of writing, there are a limited amount of systems available to the public that either provide sentiment analysis or predictions of the crypto-market. Additionally, none known that combine both sentiment and price analysis to make said predictions on the direction of the market.

Such tools are usually provided by exchanges which correspond the amount of positive and negative sentiments with a suggestion to buy and sell. These tools, however, are vague in their suggestions as they don't provide any further analysis on when the best time would be to conduct an action on the market, and simply display the number of tweets per sentiment level. A well-known cryptocurrency tracking site,Coingecko provides a basic sentiment analysis tool for their top 30 ranking cryptocurrencies tracked on the site. This tool shows the sentiment analysis of tweets from Twitter every hour for a given cryptocurrency. This is displayed as a simple pill on the page showing the ratios of positive, neutral and negative tweets. *See Appendix C for visual representation*

Related research

There has been a plentiful amount of research conducted in this problem domain. Numerous theses globally have been published in recent years on the topic of cryptocurrency market predictions and analysis, and even more, research conducted on general stock markets further back.

The thesis written by *Evita Stenqvist and Jacob Lonno* of the *KTH Royal Institute of Technology* [3] investigates the use of sentiment expressed through micro-blogging such as Twitter can have on the price fluctuations of Bitcoin. Its primary focus was creating an analyser for the sentiment of tweets more accurately *"by not only taking into account negation, but also valence, common slang and smileys"*, than that of former researchers that *"mused that accounting for negations in text may be a step in the direction of more accurate predictions."*. This would be built upon the lexicon-based sentiment analyser VADER to ascertain the overall sentiment scores were grouped into time-series for each interval from 5 minutes to 4 hours, along with the interval prices for Bitcoin. The model chosen was a naive binary classified vectors of predictions for a certain threshold to *"ultimately compare the predictions to actual historical price data"*. The results of this research suggest that a binary classification model of varying threshold over time-shifts in time-series data was "lackluster", seeing the number of predictions decreasing rapidly as the threshold changed. This research is a good basis of starting research upon, as it suggests tools such as VADER for sentiment analysis and that the use of a machine learning algorithm would be a next step in the project that would yield better more accurate results.

Another thesis written by *Pagolu, Venkata Sasank and Reddy Kamal Nayan, Panda Ganapati and Majhi, Babita* [1] on "Sentiment Analysis of Twitter Data for Predicting Stock Market Movements" 2.5 million tweets on Microsoft were extracted from Twitter, sentiment analysis and logistical regression performed on the data yielded 69.01% accuracy for a 3-day period on the increase/decrease in stock price. These results showed a "*good correlation between stock market movements and the sentiments of the public expressed in Twitter*". Using various natural language pre-processing tweets for feature extraction such as N-gram representation the sentiment from tweets were extrapolated. Both Word2vec and a random forest classifier were compared for accuracy, Word2vec being chosen over the machine learning model. Word2vec, being a group of related shallow two-layer neural network models to produce word embeddings.

A topic that reoccurs in various papers and theses is that of the use and focus of regression techniques and machine learning methods. Few implement a fully fledged neural network, the above paper attempts to use a simple network to achieve predictions of classification of sentiment for stock market movement then correlated this with historical data of prices. An article posted on "Code Project" by Intel Corporation [5] compares the accuracy of three machine learning algorithms; Random Forest, Logistic Regression and Multi-Layer Perceptron (MLP) classifiers on predicting the price fluctuations of Bitcoin with embedded price indices. Results showing *"that using the MLP classifier (a.k.a. neural networks) showed better results than logistic regression and random forest trained models"*. This assumption can be backed up by the results from a thesis posted on IEEE [6] which compares a Bayesian optimised recurrent neural network and a Long Short Term Memory (LSTM) network. Showing the LSTM model achieving *"the highest classification accuracy of 52% and a RMSE of 8%"*. With an interest in neural networks personally and with little papers utilising them for this purpose a neural network will thus be implemented, and the accuracy of one's predictions with use of sentiment analysis data analysed and discussed.

Data Collection

Twitter and Twitter API

Twitter is a micro-blogging platform that was launched in 2006 and provides it's users the ability to publish short messages of 140 characters. The messages published could be of any form, from news snippets, advertisement, or the prevalent publication of opinions which allowed a platform of extensive diversity and knowledge wealth. As of the time of writing, the message character limit was increased to 280 characters, the platform has over 300 million monthly active users and around 1 million tweets are published per day. Due to the length restriction and the primary use of the platform to express opinions Twitter is seen as a gold mine for opinion mining.

The Twitter API has an extensive range of endpoints that provide access from streaming tweets for a given hashtag, obtaining historical tweets for a given time-period and hashtag, posting tweets on a users account and to change settings on a user account with authentication. The exhaustive range of features provided by these endpoints makes data collection from Twitter straight forward as one can target a specific endpoint for the required data. Due to Twitter being the target for opinion mining within this project the Twitter API will ultimately need to be utilised. This can either be used for the gathering of historical tweets or streaming current tweets for the #Bitcoin hashtag.

There are, however, limitations and rate limits imposed on users of the API. Twitter employs a tiering system for the API - Standard, Premium and Enterprise tiers, each of which provides different amounts of access for data collection. If the API were used to capture historical data for a span of 3 months, each tier is allowed to obtain varying amounts of data for different durations; [7]

- A Standard user would be able to capture 100 recent tweets for the past 7 days
- A Premium user would be allowed to capture up to 500 tweets per request for a 30-day span and will have access to a full-archive search to query up to 100 tweets per request for a given time period, with a 50 request limit per month
- An Enterprise user would be able to capture up to 500 tweets per unlimited requests for a 30-day span and will be able to query the full-archive of tweets for a given hashtag up to 2000 tweets per unlimited amount of requests for a given time period

Each tier has individual costs while the standard user negating this as a basic tier. Due to only being elegable for the Premium tier for educational purposes, historical data gathering will be limited to

100 tweets per request with a limitation of 50 requests per month. Furthermore streaming tweets is an Enterprise feature which rules out the the Twitter API for use of streaming current real-time data [8].

Tweepy Python Package

Tweepy is a python package for accessing the Twitter API. It fundamentally accomplishes the same means if one to conduct a GET request to the Twitter API, except it simplifies this into a simple to use API that is easier to implement and automate in python [9]. Consequently, it builds upon the existing Twitter API to provide features such as automated streaming of provided hashtags to the API. It realises this by initialising a listener instance for a provided set of API credentials, handling authentication, connections, creating and destroying sessions. Due to Twitter's streaming API being only available to Enterprise users [7], using Tweepy to stream data for a given hashtag will provide the real-time data needed. The streaming of current data by Tweepy is accomplished by setting up a stream which listens for new data for a given hashtag, which bypasses the need for the Enterprise tweet tracker provided by the Twitter API.

Sentiment Analysis

In short, sentiment analysis is the process and discovery of computationally identifying and categorising the underlining opinions and subjectivity expressed in written language. This process determines the writer's attitude towards a particular topic as either being positive, neutral or negative in terms of opinion, known as polarity classification.

Natural Language Processing

Polarity classification is the focus of sentiment analysis and is a well-known problem in natural language processing that has had significant attention by researchers in recent years [1][3][6][10]. Traditional approaches to this have usually been classified to dictionary-based approaches that use a pre-constructed sentiment lexicons such as VADER or usually confined to machine learning approaches. The later requires an extensive amount of natural language pre-processing to extrapolate vectors and features from given text, this is then fed into a machine learning classifier which attempts to categorise words to a level of sentiment polarity. Natural language pre-processing techniques that would be required for this approach would consist of;

- Tokenisation: The act of splitting a stream of text into smaller units of typographical tokens which isolate unneeded punctuation.
- Removal of domain specific expressions that are not part of general purpose English tokenisers - a particular problem with the nature of the language used in tweets, with @-mentions and #-hashtags
- Stopword removal: Are commonly used words (such as "the","in","a") that provide no meaning to the sentiment of a given text
- Stemming: Is used to replace words with common suffixes and prefixes, as in "go" and "goes" fundamentally convey the same meaning. A stemmer will replace such words with their reduced counterparts

- Term Probability Identification and Feature Extraction: This is a process that involves identifying the most frequently used words in a given text, by using a probability type approach on a pre-defined dataset which classifies a range of texts as with overall negative or positive a machine learning algorithm is trained to classify these accordingly.

The former, seen and has been proven to provide higher accuracy than traditional machine learning approaches [11], and need little pre-proeccesing conducted on the data as words have a pre-defined sentiment classification in a provided lexicon. Although these lexicons can be complex to create, they generally require little resources to use and add to.

Valence Aware Dictionary and sEntiment Reasoning

VADER is a combined lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains. It is capable of detecting the polarity of a given text - positivity, neurality, and negativity [12]. VADER uses a human-centric approach to sentiment analysis, combining qualitative analysis and empirical validation by using human raters to rate level of sentiment for words in its lexicon. Vader also has emoticon support which maps these colloquailisms have pre-defined intensities in its lexicon, which makes VADER specifically suitable for the social media domain where the used of emoticons, utf-8 emojis and slang such as "Lol" and "Yolo" are prevalent within text. Additionally, VADER is provided as a lexicon and a python library under the MIT license, thus means that it is open-source software. This means that the lexicon can be altered and added to making it able to being tailored to specific topic domains.

VADER was constructed by examining and extracting features from three pre-existing well-established and human-validated sentiment lexicons [12] - (LIWC) Linguistic Inquiry and Word Count, (ANEW) Affective Norms for English Words, and (GI) General Inquirer. This is supplemented with additional lexicon features *"commonly used to express sentiment in social media text (emoticons, acronyms and slang)"* [12] and uses "wisdom-of-the-crowd" approach [13] to establish a point of estimations of sentiment valance for each lexical feature candidate. This was evaluated for the impact of grammatical and syntactical rules and 7,500+ lexical features, with mean valence *"¡¿ zero, and SD ¡= 2.5"* as a human-validated "gold-standard" sentiment lexicon. [12]*Section 3.1*

VADER is seen as a "Gold Standard" for sentiment analysis, in the paper for VADER, [12] *A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*, it was compared against 11 other *"highly regarded sentiment analysis tools/techniques on a corpus of over 4.2K tweets"* for polarity classification across 4 domains. Results showing VADER, across Social media text, Amazon reviews, movie reviews and Newspaper editorials, consistently outperforming other sentiment analysis tools and techniques showing a particular trend in performing significantly higher on analysis of sentiment in tweets. [12] *Section 4: Results*

Neural Networks

A neural network is a set of perceptrons modelled loosely after the human brain that is designed to recognise patterns in whatever domain it is intended to be trained on. A neural network can consist of multiple machine perceptrons or clustering layers in a large mesh network and the patterns they recognise are numerical which are contained in vectors. Pre-processed data, confined and processed into pre-defined vector labels, are used to teach a neural network for a given task. While this differs from how an algorithm is coded to a particular task, neural networks cannot be programmed directly for the task. The requirement is for them to learn from the information by use of different learning strategies; [14][15]

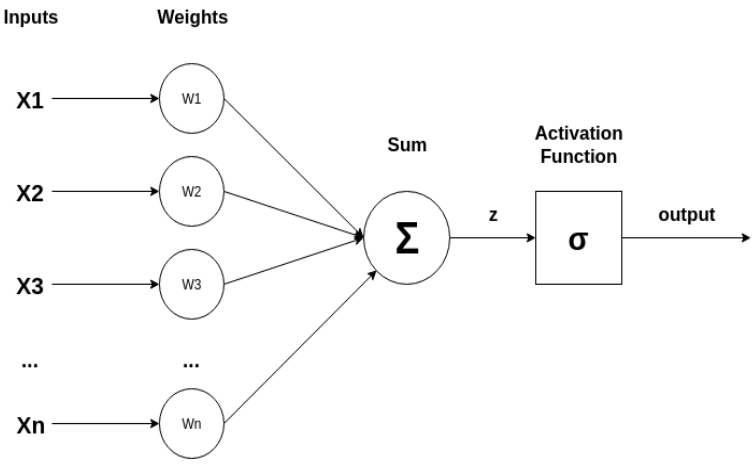

Figure 1: Basic perceptron layout

- Supervised learning: Simplest of the learning forms, where a dataset have been labeled which indicate the correct classified data. The input data is learned upon until the desired result of the label is reached [16]
- Unsupervised learning: Is training the with a dataset without labels to learn from. The neural network analyses the dataset with a cost function which tells the neural network how far off target a prediction was. The neural network then adjusts input weights in attempt to increase accuracy. [15]
- Reinforced learning: The neural network is reinforced with positive results and punished for negative results causing a network to learn over iterations.

Recurrent Neural Network (RNN)

The type of neural network that is of focus for this project will be that of a Long-Short Term Memory (LSTM), however, it is important to understand how this is an extension of a Recurrent Neural Network (RNN) and how the underlying network works.

Recurrent Neural Networks (RNN) are a robust and powerful type of neural network and is considered to be among the most encouraging algorithms for use of classification, due to the fact of having internal memory. RNNs are designed to recognise patterns in sequences of presented data or most suitably, time-series data, genomes, handwriting and stock market data. Although RNNs were conceptualised and invented back in the 1980s [17] they've only really shown their potential in recent years, with the increase of computational power due to the level of sequencing and internal memory store to retrain. Due to this 'internal' memory loop, RNNs are able to remember data and adjust neurons based on failures and alternating parameters. The way this is accomplished, knowing how a standard neural network such as a feed-forward network, should initially be understood. [18]

A standard, feed-forward neural network has a single data flow with an input layer, through hidden computational layers, to an output layer. Therefore any node in the network will never see the same data again. However, in an RNN data is cycled through a loop over the same node, thus two inputs into the perceptron. Decisions are influenced by previous data that it has previously learned from if any, which in turn affects output and the weights of the network. [19]

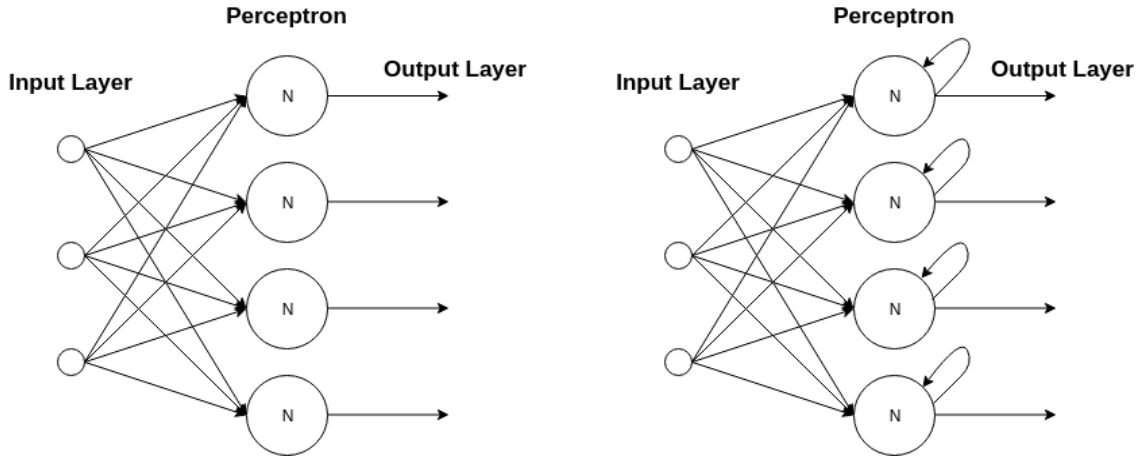

Figure 2: Feed-forward network (left) vs Recurrent Neural network (right)

The act of tweaking weights to alter the processing of the next iteration of data in an RNN is called backpropagation, which in short means going back through the network to find the partial derivatives of the error with respect to the weights after output has occurred. Along with gradient descent, an algorithm that adjusts the weights up or down depending on which would reduce the error. There are however a few obstacles of RNNs;

- Exploding Gradients: Is when gradient decsent assigns high importance to the weights. As in the algorithm assigns a ridiculously high or low value for the weights on iteration which can cause overlow and result in NaN values [20]
- Vanishing Gradients: Is when the values of a gradient are small enough that weights cannot be altered and the model stops learning. [21]

These issues are overcome by the concept of Long-Short Term Memory neural networks, coined by *Sepp Hochreiter and Juergen Schmidhuber, 1997* [22].

Long-Short Term Memory (LSTM)

LSTMs are an extension of recurrent neural networks capable of learning long-term dependancies and were conceptualised by *Sepp Hochreiter and Juergen Schmidhuber, 1997* [22]. LSTMs were explicity designed to avoid long-term dependancy problems such as exploding and vanishing gradients. As they are an extension of RNNs they operating in almost the exact same manner, but stores the actual gradients and weights in memory which allows for LSTMs to read, write and alter the values. A way of explaining how this works is seeing the memory block as a gated cell, where 'gated' is that the cell decides whether or not to store or alter data in it's memory based input data and the importance assigned to it. In a sense it learns over time of which values and data is important.

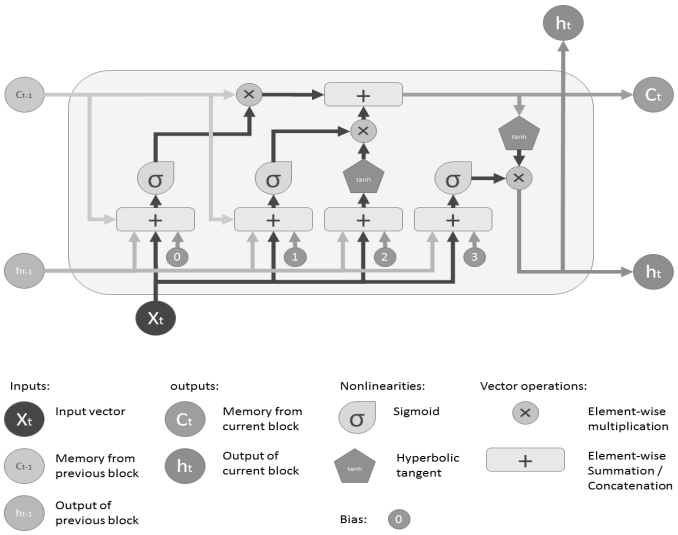

Figure 3: A conceptual design of an LSTM cell bank - from Medium article by Shi Yan: Understanding LSTM and its diagrams[23]

The network takes in three initial inputs, input of current time step, output from the previous LSTM unit if any, and the memory of the previous unit. Outputs, H_t - output of current network, and C_t - the memory of the current unit. [23]

The various steps of the network decide what information is thrown away from the cell state, through use of a 'forget gate' which is influencted by the calculations of sigmod memory gates which influence how much of old and new memory is used C_{t_1}, H_{t-1} and X_t, and merged together based upon importance. The section of the cell that controls the outflow memory H_t and C_t determines how much of the new memory should be used by the next LSTM unit. *For a more in-detailed explanation of exactly how the calculations are made see* [22],[23] and [24].

As mentioned in the formost section of related work the use of an LSTM network would be optimal for the given problem domain over the use of machine learning algorithms, for time-series data. As detailed above, LSTMs are widley used for time-series data forecasting due to being able to remember previous data and weights over long sequence spans[22][25]. The flexability of LSTMs such as many-to-many models, useful *"to predict multiple future time steps at once given all the previous inputs"* due to use of look-back windows and control of multiple 3D input parameters.[25]

Keras and TensorFlow

TensorFlow is an open-source numerical math computational library framework for dataflow differentiable programming, primarily used for machine and deep learning applications such as neural networks. TensorFlow bundles various machine learning and deep learning models and algorithms into one package for the Python language, but executes as byte code executed in C++ for performance. TensorFlow provides a range of conveniences to developers for the types of algorithms it supports such as debugging models and modifying graph operations separately instead of constructing and evaluating all at once, and compatibility to execute on CPUs or GPUs [26]. However, TensorFlow's implementation and API, although provides an abstraction for development for machine and deep learning algorithms and simplifies implementation, it isn't all too friendly to programmers to use, especially new developers to the field of machine and deep learning. This is were the Keras API comes in.

Keras is a high-level built to run atop of deep learning libraries such as Tensorflow and Theanos - another deep learning library similar to Tensorflow. It is designed to further simplify the use and application of such deep learning libraries thus making implementing a neural network and similar supported algorithms friendlier to developers working in Python. It accomplishes this by being a modular API; neural layers, cost functions, optimisers, activation functions, and regularisation schemes are all standalone features of the API that can be combined to create functional or sequential models. Due to being a high-level API for more refined and easier development of deep learning libraries it does not provide these low-level operations and algorithms; Keras relies on a back-end engine such as TensorFlow and supports a wide range of others.

Optimisers

There are three distinct optimisers used for LSTM networks; ADAgrad optimizer, RMSprop, Adam. The role of an optimiser All three of which is a type of Stochastic Gradient Descent, which θ (weights of LSTM) is changed according to the gradient of the loss with respect to θ. Where α is the learning rate and L is the gradient loss. [27]

$$\theta_{t+1} = \theta_t - \alpha\delta L(\theta_t)$$

This is primarily used in recurrent LSTM neural networks to adjust weights up or down depending on which would reduce the error, *see RNN section for non LSTM limitations.* The concept of using momentum μ in stochastic gradient decent helps to negate significant convergance and divergance during calculation of the weights and dampens the oscillation, by increasing the speed of the learning rate upon each iteration. [28]

$$\theta_{t+1} = \theta_t + v_{t+1}$$

where

$$v_{t+1} = \mu v_t - \alpha\delta L(\theta_t)$$

[28]

- Adagrad (Adaptive Gradient): Is a method for adaptive rate learning through adaptively changing the learning parameters. This involves performing larger updates for infrequent parameters and smaller updates for frequent parameters. This algorithm fundamentally eliminates the need to manually tune the learning rate of the neural network, and is well suited with sparse data in a large scale network. [28]

$$\theta_{t+1} = \theta_t + v_{t+1}\frac{\eta}{\sqrt{G_t + \epsilon}} \cdot g_t$$

(G_t is the sum of the squares of the past gradients to θ)

- RMSProp (Root Mean Square Propagation): Aims to resolve Adagrads radically diminishing learning rates by using a moving average of the squared gradient. Thus utilises the magnitude of the recent gradient decsent to normalise it, and gets adjusted automatically by choosing different learning rate for each parameter. [29]

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{(1-\gamma)g_{t-1}^2 + \gamma g_t + \epsilon}} \cdot g_t$$

(γ - decay that takes value from 0-1. g_t - moving average of squared gradients)

[30]

- Adam (Adaptive Moment Estimation): Also aims to resolve Adagrads diminishing learning rates, by calculates the adaptive learning rate for each parameter. Being one of the most popular gradient decsent optimisation algorithms, it estimates from the 1st and 2nd moments of gradients. Adam implements the exponential moving average of the gradients to scale the learning rate of the network, and keeps an average of past gradients. [31]

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

The algorithm updates the moving averages of the gradient (m_t) and the squared gradient (v_t) which are the estimates of the 1st and 2nd moments respectively. The hyperparameters β_1 and β_2 control the decay rates of the moving averages. These are initialised as 0 as a biased estimations for the initial timesteps, but an become bias-corrected by counteracting them with;

$$\vec{m}_t = \frac{m_t}{1 - \beta_1^t}$$

and

$$\vec{v_t} = \frac{v_t}{1 - \beta_2^t}$$

Thus the final formula for the Adam optimiser is;

$$\theta_{t+1} = \theta_t - \frac{\eta \vec{m}_t}{\sqrt{\vec{v_t} + \epsilon}}$$

Diederik P. Kingma, Jimmy Lei Ba - Adam: A method for Stochastic Optimization[30]

Machine Learning

Naive Bayes

To get an understanding of both how probability works and how a neural network will predict the next hour value based on the concepts of probability, using a well-established probability algorithm will aid in this understanding.

Bayes theorem works on conditional probability and is the probability of how often an event will happen given that that event has already occurred. There are numerous variations of the theorem such as Multinomial, which supports categorical features where each conforms to a multinomial distribution, and Gaussian naive Bayes, which support continuous-valued features each of which conforming to a Gaussian (normal) distribution. The classical multinomial Bayes theorem is defined as; [32]

$$P(H \cap A) = \frac{P(A \cap H) * P(H)}{P(A)}$$

and incase H and A are independant

$$P(H \cap A) = P(H) => P(H \cap A) = P(H)P(A)$$

where:

- $P(H)$ is the probability of hypothesis being true
- $P(A)$ is the probability of evidence
- $P(A \cap H)$ is the probability of the evidence such that the hypothesis is true
- $P(H \cap A)$ is the probability of the hypothesis given the occurance of evidence of the probability

The naive approach assumes the features that are used in the model are independent of one another, such that, changing the value of a feature doesn't directly influence the value of the other features used in the model. When such features are independent, the Bayes algorthim can be expanded:

$$P(H \cap A) = \frac{P(A \cap H) * P(H)}{P(A)}$$

Becomes

$$P(H \cap A_1 ... A_n) = \frac{P(A_1 \cap H) * P(A_2 \cap H) ... * P(A_n \cap H) * P(H)}{P(A_1) * P(A_2) ... * P(A_n)}$$

$$Probability\ of\ Outcome \cap Evidence = \frac{Probability\ of\ Likelihood\ of\ evidence * Prior}{Probability\ of\ Evidence}$$

The naive Bayes approach has many applications, especially for the topic of this project in classifying the probability occurance of the next price. Although it is a robust algorithm is does have its drawbacks which make it not as suitable as a neural network for the given need of this project. The naive Bayes trap is an issue that may occur due to the size of dataset that will be used. There are however other scenarios this algorithm could be used such as classification of spam data.[32]

Solution Approach

This section will outline the solution intended to solve the problem that the problem statement identifies with justification and reference to the research conducted in the litrature review. This will lay out the development process for the project and will tools and technologies will be explained for the particular use case in this project.

Data gathering

This will be the part of the system that will gather price data and tweets from relevent sources, twitter and cryptocurrency exchanges.

Price data

Historical price data can be collected in a number methods, one being that of the exchange APIs, another through a historical price tracker who creates a CSV consisting of all prior historical data. Both have their merits and reliability for granting the needed data, however, a historical tracker who has been tracking the price every hour since the start of Bitcoin would be the better option. This is due to a couple of factors, the data in some historical trackers are an average unbiased price for Bitcoin - they track the price of all or a select few exchanges and average the hourly price. Whereas if the historical data was obtained directly from an exchange this would be biased and might not represent the true price of the currency, and thus would need averaging with other hourly prices from other exchanges. By using a historical tracker all the data is unbiased and averaged and readily available and doesn't require any requests to an API or coding needed to process data.

Live price data can be collected through the same methods, a historical price tracker and an exchange API. However, this doesn't work the same way, unfortunately, a historical price tracker isn't updated as frequently as exchange APIs thus wouldn't provide on the hour accurate data. Therefore exchange APIs should be utilised in this case and multiple to provide an unbiased average for the hourly price.

Tweets

Historical tweets can be obtained through the Twitter API, and however is not a feature of the Tweepy package - *not mentioned or method on official Tweepy Documentation* [33]. The Twitter API, as explained in the Litrature review, allows for historical tweets to be extracted from the platform, 100 per request and a maximum of 50 requests per month. This proposes an issue with not providing enough data, where sentiment will need to be calculated per hour. Simply put, for a year of hourly price data there will be 9050 records. Therefore the equivilent will be needed for sentiment, however the sentiment will be the average the sentiment per hour of tweets. Using one request, 100 tweets per hour, per hour, 905,000 tweets will need to be extracted to provide the data needed. A solution to this issue could be to use and create multiple accounts and manually extract data from the API and merge. Another option is the pay for the data from 3rd party companies whom have access to the Enterprise API and can pull more data, 2000 per request
cite7[8]. Due to price for data of these 3rd parties the former could be a suitable, but more time consuming option.

Live tweets can be collected by two methods from Twitter, from the Twitter API and using Twitter Python package such as Tweepy, detailed in the Literature review. Additionally, the limitations of the Twitter API are also discussed in the review which states how the Twitter API has a tiering system: Standard, Premium and Enterprise. Each tier has different levels of access to the API and can extract

a different amount of data from the platform. Thus concluding the section in the Literature review, the Twitter API will not be used for the extraction and streaming of live tweets due to it being restricted to Enterprise users. Therefore, Tweepy will be used to set up a looping authenticated streaming solution with the Twitter API which will allow the access of current recurring data.

Spam Filtering

This part of the system will aim to detect whether or not the steamed and/or the historical tweet is spam - unwanted tweets that serve no purpose in determining opinion of the public. These types of tweets can be from advertisement - usually labeled with *#Airdrop* and can contain *"tickets here" and "Token Sale"*, to job advertisments - usually containing word such as *Firm, hire, hiring, jobs and careers.*

Language Detection

Twitter provides this, but non-basic-latin characters can get through in tweets

Solution Summary

Requirements

Data flow Overview

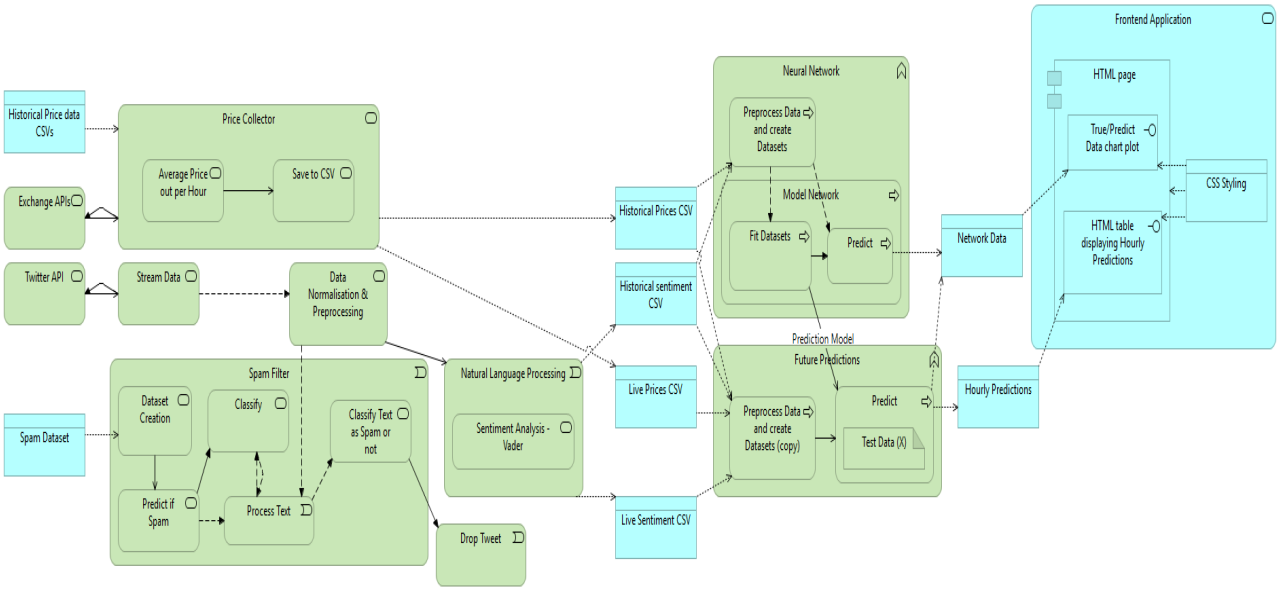

Figure 3: A conceptual design of an LSTM cell bank - from Medium article by Shi Yan: Understanding LSTM and its diagrams[23]

Packages, Tools and Techniques

System Design and Implementation

Data collection

Price Time-series Data

Historical data of Bitcoin prices can be obtained through may means,

Data processing

Preprocessing

Tweet Filtering

Text Cleaning

Ngram based Language detection filtering

Spam Filtering

Tweet Processing

Naive Bayes model

Sentiment Analysis

VADER

Recurrent Neural Network - LSTM

Training and Testing Model

Dropouts?

Scoring and Validation

Loss?

Future Prediction Forecasting

Testing: Verification and Reflection

Discussion: Contribution and Reflection

Limitations

Conclusion and Future Improvements

Conclusion

Future Improvements

Shifting the intial data by and hour and sequencing over previous data - will also allow proper use of look-back windows

Another could be to predict the hour of sentiment and create a threshold for it.

References

[1] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of twitter data for predicting stock market movements," in *2016 international conference on signal processing, communication, power and embedded system (SCOPES)*, IEEE, 2016, pp. 1345–1350. [Online]. Available: https://arxiv.org/pdf/1610.09225.pdf.

[2] N. Indera, I. Yassin, A. Zabidi, and Z. Rizman, "Non-linear autoregressive with exogeneous input (narx) bitcoin price prediction model using pso-optimized parameters and moving average technical indicators," in *Journal of Fundamental and Applied Sciences. Vol.35, No.35*, University of El Oued, 2017, pp. 791–808. [Online]. Available: https://www.ajol.info/index.php/jfas/article/viewFile/165614/155073.

[3] J. L. Evita Stenqvist, "Predicting bitcoin price fluctuation with twitter sentiment analysis," Diva, 2017. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1110776/FULLTEXT01.pdf.

[4] O. G. Yalcin, "Predict tomorrows bitcoin (btc) price with recurrent neural networks," Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/using-recurrent-neural-networks-to-predict-bitcoin-btc-prices-c4ff70f9f3e4.

[5] Intel-Corporation, "Stock predictions through news sentiment analysis," Code Project, 2017. [Online]. Available: https://www.codeproject.com/Articles/1201444/Stock-Predictions-through-News-Sentiment-Analysis.

[6] S. C. Sean McNally Jason Roche, "Predicting the price of bitcoin using machine learning," in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, IEEE, 2018, pp. 344–347. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8374483.

[7] Twitter, "Search tweets," Twitter Developers, 2018. [Online]. Available: https://developer.twitter.com/en/docs/tweets/search/overview.

[8] ——, "Consuming streaming data," Twitter Developers, 2018. [Online]. Available: https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.html.

[9] J. Roesslein, "Streaming with tweepy," Tweepy, 2009. [Online]. Available: http://docs.tweepy.org/en/v3.4.0/streaming_how_to.html.

[10] S. N. Mehrnoush Shamsfard, "Using linked data for polarity classification of patients experiences," in *Journal of Biomedical Informatics*, Elsevier, 2015, pp. 6–19. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046415001276.

[11] L. P. T. Chedia Dhaoui Cynthia M. Webster, "Social media sentiment analysis: Lexicon versus machine learning," in *Journal of Consumer Marketing, Volume 34. Issue 6*, Emerald Insight, 2017. [Online]. Available: https://www.emeraldinsight.com/doi/pdfplus/10.1108/JCM-03-2017-2141.

[12] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, Ann Arbor, MI, 2014. [Online]. Available: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/download/8109/8122.

[13] W. Kenton, "Wisdom of crowds," Investopedia, 2018. [Online]. Available: https://www.investopedia.com/terms/w/wisdom-crowds.asp.

[14] Skymind, "A beginner's guide to neural networks and deep learning," in *A.I. Wiki*, Skymind, 2018. [Online]. Available: https://skymind.ai/wiki/neural-network.

[15] J. DeMuro, "What is a neural network," in *World of tech*, techradar, 2018. [Online]. Available: https://www.techradar.com/uk/news/what-is-a-neural-network.

[16] F. Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, NIPS Proceedings, 2009, pp. 1033–1040. [Online]. Available: http://papers.nips.cc/paper/3448-supervised-dictionary-learning.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, 1985. [Online]. Available: https://apps.dtic.mil/docs/citations/ADA164453.

[18] Skymind, "A beginner's guide to lstms and recurrent neural networks," in *A.I. Wiki*, Skymind, 2018. [Online]. Available: https://skymind.ai/wiki/lstm.

[19] N. Donges, "Recurrent neural networks and lstm," Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5.

[20] P. Jason Brownlee, "A gentle introduction to exploding gradients in neural networks," Machine Larning Mastery, 2017. [Online]. Available: https://machinelearningmastery.com/exploding-gradients-in-neural-networks/.

[21] S. D. S. Team, "Recurrent neural networks (rnn) - the vanishing gradient problem," Super Data Science, 2018. [Online]. Available: https://www.superdatascience.com/blogs/recurrent-neural-networks-rnn-the-vanishing-gradient-problem.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," in *Neural computation, Volume 9. 8*, MIT Press, 1997, pp. 1735–1780. [Online]. Available: https://www.bioinf.jku.at/publications/older/2604.pdf.

[23] S. Yan, "Understanding lstm and its diagrams," Medium, Mar 13, 2016. [Online]. Available: https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714.

[24] C. Olah, "Understanding lstm networks," 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs.

[25] R. Kompella, "Using lstms to forecast time-series," Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/using-lstms-to-forecast-time-series-4ab688386b1f.

[26] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th Symposium on Operating Systems Design and Implementation 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

[27] Stanford, "Optimization: Stochastic gradient descent," in *UFLDL Tutorial*. [Online]. Available: http://deeplearning.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent.

[28] R. Mitra, "What are differences between update rules like adadelta, rmsprop, adagrad, and adam," Quora, 2016. [Online]. Available: https://www.quora.com/What-are-differences-between-update-rules-like-AdaDelta-RMSProp-AdaGrad-and-AdaM.

[29] M. C. Mukkamala and M. Hein, "Variants of rmsprop and adagrad with logarithmic regret bounds," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR.org, 2017, pp. 2545–2553. [Online]. Available: https://arxiv.org/pdf/1706.05507.pdf.

[30] R. Khandelwal, "Overview of different optimizers for neural networks," Medium, 2019. [Online]. Available: https://medium.com/datadriveninvestor/overview-of-different-optimizers-for-neural-networks-e0ed119440c3.

[31] J. L. B. Diederik P. Kingma, "Adam: A method for stochastic optimization," in *arXiv preprint arXiv:1412.6980*, arXiv, 2014. [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf.

[32] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, pp. 41–46. [Online]. Available: https://www.cc.gatech.edu/~isbell/reading/papers/Rish.pdf.

[33] J. Roesslein, "Tweepy documentation," 2009. [Online]. Available: http://docs.tweepy.org/en/v3.5.0/.

Appendices

Appendix A - Project Initiation Document

Displayed on the following pages below.

Individual Project (CS3IP16)

Department of Computer Science
University of Reading

Project Initiation Document

PID Sign-Off

Student No.	**24005432**
Student Name	**Andrew Sotheran**
Email	**andrew.sotheran@student.reading.ac.uk**
Degree programme (BSc CS/BSc IT)	**BSc CS**
Supervisor Name	**Kenneth Boness**
Supervisor Signature	
Date	

SECTION 1 – General Information

Project Identification

1.1	**Project ID** (as in handbook)
	N/A
1.2	**Project Title**
	Cryptocurrency market and value prediction tracking
1.3	**Briefly describe the main purpose of the project in no more than 25 words**
	To provide a means to predict the value of cryptocurrencies that will aid in investor decision making in investment of the market

Student Identification

1.4	**Student Name(s), Course, Email address(s)** e.g. Anne Other, BSc CS, a.other@student.reading.ac.uk
	Andrew William Sotheran BSc CS Andrew.sotheran@student.reading.ac.uk

Supervisor Identification

1.5	**Primary Supervisor Name, Email address** e.g. Prof Anne Other, a.other@reading.ac.uk
1.6	**Secondary Supervisor Name, Email address** Only fill in this section if a secondary supervisor has been assigned to your project

Company Partner (only complete if there is a company involved)

1.7	**Company Name**
	N/A
1.8	**Company Address**
	N/A
1.9	**Name, email and phone number of Company Supervisor or Primary Contact**
	N/A

SECTION 2 – Project Description

2.1 **Summarise the background research for the project in about 400 words. You must include references in this section but don't count them in the word count.**

To create a tool that aims to predict the price of cryptocurrencies that aids in investor decisions. Research will need to be conducted into the following topics that surround data mining, machine learning and artificial neural networks.

This research will consist along the lines of;
Natural Language processing and analysis – To analyse and process fed in data gathered through RSS data feeds and social media feeds, through the underlying tasks of Natural language processing. Content categorisation (search and indexing, duplication detection), Topic discovery and modelling (Obtain meanings and themes within the data and perform analytic techniques), sentiment and semantic analysis (which will identify the mood and opinions within the data), summariser (to summarise a block of text and disregard the rest).

Machine learning algorithms: The three types of machine learning (Supervised, Unsupervised and Reinforced)
The types of common algorithms used, each of these will be researched to identify the most suitable for this project and only one will be used: (Linear Regression, Logistic Regression, Decision Tree, SVM, Naive Bayes, kNN, K-Means, Random Forest, Dimensionality Reduction Algorithms, Gradient Boosting algorithms (GBM, XGBoost, LightGBM, CatBoost).

Artificial Neural Networks: To identify the drawbacks and benefits of using them or other computational models within machine learning. Recurrent Neural networks and 3rd generation Neural Networks.

Data mining: To investigate the different techniques and algorithms used (Same as the ones listed above for machine learning including C4.5, Apriori, EM, PageRanks, AdaBoost and CART) these will be researched and the most appropriate identified.

To investigate techniques: for storing and processing large amount of data, such as Hadoop, Elasticsearch utilities, Graphing and data modelling and visualisation.

To identify appropriate libraries for python or C for each of the topics above to aid in the creation of this project. Libraries such as:
Natural Language Toolkit (NLTK) – python
Pandas - python
Sklearn - python
Numpy – python - scientific computation for working with arrays
Matplotlib - python - data visualisation

Investigate into types of databases. Sql and nosql for a storage medium between receiving data and feeding it into the machine learning algorithm.
Investigate into the use of REST API and other web-service based technologies (GRPC, Elasticsearch)
Investigate into frameworks for the thin client, such as Angular vs React, Nodejs, Leafelt.js, charts.js
Additionally Web scraping may be needed if certain website that don't either have an API or JSON for the data needed.

https://www.sas.com/en_gb/insights/analytics/what-is-natural-language-processing-nlp.html
https://blog.algorithmia.com/introduction-natural-language-processing-nlp/
https://gerardnico.com/data_mining/algorithm
https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/
https://www.kdnuggets.com/2015/05/top-10-data-mining-algorithms-explained.html
https://www.datasciencecentral.com/profiles/blogs/artificial-neural-network-ann-in-machine-learning
http://scikit-learn.org/stable/index.html
https://grpc.io/docs/

2.2	**Summarise the project objectives and outputs in about 400 words.** These objectives and outputs should appear as tasks, milestones and deliverables in your project plan. In general, an objective is something you can do and an output is something you produce – one leads to the other.
	To produce a thin web client that provides a dashboard that provides tangible and useful information to users such as; Their current price (Updated every 5 minutes), exchange rates, network hashrates, historical price data. It will also display statistics about sentiment analysis conducted on social media about the currency, graphical predictions on what the price may be, in a given time, and will also compare this to other currencies for aid in investment. To produce significant research into the topics in and around data mining, machine learning and Artificial Neural network and the underlying tasks and algorithms used, the efficiency, drawbacks and advantages of each to identify the most suitable for the use in this project. To produce a system that analyses a data set obtained through social media feeds and posts on news sites regarding crypto currencies. It should perform sentiment analysis using Natural Language processing and analysis techniques to identify features and identifies the type of sentiment in the data and categorises it for machine learning. To utilise machine learning techniques and algorithms to produce a system that learns from historical data to predict to an extent the possible future price of a given currency. To compare this with the use of an Artificial Neural Network and to analyse the drawbacks of both.
2.3	**Initial project specification - list key features and functions of your finished project.** Remember that a specification should not usually propose the solution. For example, your project may require open source datasets so add that to the specification but don't state how that data-link will be achieved – that comes later.
	The finished project should provide a thin client single page application. This will provide a means to users the ability to view various statistics on crypto currencies on a dashboard that incorporates text analysis through natural language analysis, and will utilise various machine learning and data mining techniques to provide price predictions to the users. The nature and level of this will depend on the research conducted into the areas of data mining, machine learning, natural language processing and artificial neural networks, along with the algorithms used. The data set will be created from scratch for this project as it will require the gathering of data from numerous sources and performing text analysis on them to for the data needed. Data sets for the characteristic and data for the currencies can be obtained from pre-existing data sets such as: https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory https://www.kaggle.com/jessevent/all-crypto-currencies Web scraping may be included if certain news/social media websites do not provide an API or RSS feed for the analysis engine to perform text analysis on Additionally, there will be a server between the analysis/prediction engine and the thin client that will maintain a database, either SQL or NoSQL, that will hold statistics about the currencies and data about the price predictions about the currencies. It will not hold any of the data used in the analysis engine, as this database will only hold data available to the end users.

2.4	**Describe the social, legal and ethical issues that apply to your project. Does your project require ethical approval? (If your project requires a questionnaire/interview for conducting research and/or collecting data, you will need to apply for an ethical approval)**
	The project will not be handling any user related data, therefore it does not need ethical approval.
2.5	**Identify and lists the items you expect to need to purchase for your project. Specify the cost (include VAT and shipping if known) of each item as well as the supplier.** e.g. item 1 name, supplier, cost
	None Needed
2.6	**State whether you need access to specific resources within the department or the University e.g. special devices and workshop**
	Possibly a server to host the database and analysis engine on to perform the computation necessary, and a server to host the thin client.

SECTION 3 – Project Plan

3.1	**Project Plan** Split your project work into sections/categories/phases and add tasks for each of these sections. It is likely that the high-level objectives you identified in section 2.2 become sections here. The outputs from section 2.2 should appear in the Outputs column here. Remember to include tasks for your project presentation, project demos, producing your poster, and writing up your report.

Task No.	Task description	Effort (weeks)	Outputs
1	**Background Research**		
1.1	Investigate into RPC frameworks and REST APIs	0.3	To identify the type of API/RPC framework that would be most suitable
1.2	Research into Natural Language processing and analysis techniques	0.5	To get an understanding of how NLP works and how it could be used
1.3	Research into the use of machine learning – types and algorithms	0.5	To grasp how ML paradigms work and how this project will use it
1.4	Research into the application of Neural Networks – drawbacks and advantages of using them	0.3	To identify whether there will be a need for a neural network or ML paradigms can be used instead
1.5	Research techniques for storing and processing large amount of data, such as Hadoop, spark or Elasticsearch utilities.	1	To understand the uses, application and whether the use of these are more viable solution than standard ML practices
1.6	Identify appropriate libraries for data modelling and visualisation, NLP and Machine Learning	1	To identify what libraries will aid in the construction of this project
1.7	Investigate into frameworks for the front-end thin clients	0.3	To identify what frameworks the thin client should be used with, along with drawbacks and advantages
1.8	Research web scraping techniques	0.3	To understand the application of these techniques and learn how to apply them
2	**Analysis and design**		
2.1	Resolve issues discovered by background research	0.2	...
2.2	Identify limitations discovered from research and what is not feasible	0.1	…
2.3	UML Diagrams/ XUML	0.2	
2.4	Wire frames for frontend	0.1	
2.5	Data Flow	0.1	
2.6	User Flow	0.1	
3	**Develop prototype**		
3.1	Develop thin client	2	
3.2	Develop analysis Engine	4	
3.3	Develop Prediction Engine	3	
3.4	Develop Unit tests	2	
4	**Testing, evaluation/validation**		
4.1	Unit testing	1	
4.2	Acceptance Testing	0.8	
4.3	User testing	0.8	
5	**Assessments**		
5.1	write-up project report	2	Project Report
5.2	produce poster	0.5	Poster
5.3	Log book	0.5	

TOTAL	**Sum of total effort in weeks**	**21.9**	

SECTION 4 - Time Plan for the proposed Project work

For each task identified in 3.1, please *shade* the weeks when you'll be working on that task. You should also mark target milestones, outputs and key decision points. To shade a cell in MS Word, move the mouse to the top left of cell until the curser becomes an arrow pointing up, left click to select the cell and then right click and select 'borders and shading'. Under the shading tab pick an appropriate grey colour and click ok.

	START DATE: 10/2018 **<enter the project start date here>** **Project Weeks**												
Project stage	0-3	3-6	6-9	9-12	12-15	15-18	18-21	21-24	24-27	27-30	30-33	33-36	36-39
1 Background Research													
Investigate into RPC frameworks and REST APIs													
Research into Natural Language processing													
Research into the use of machine learning –													
Research into the application of Neural													
Research techniques for storing and													
Identify appropriate libraries for data													
Investigate into frameworks for the front-													
Research web scraping techniques													
2 Analysis/Design													
Resolve issues discovered by background													
Identify limitations discovered from													
UML Diagrams/ XUML													
Wire frames for frontend													
Data Flow													
User Flow													

3 Develop prototype.													
Develop thin client													
Develop analysis Engine													
Develop Prediction Engine													
Develop Unit tests													
4 Testing, evaluation/validation													
Unit testing													
Acceptance Testing													
User testing													
5 Assessments													
write-up project report													
produce poster													
Log book													

RISK ASSESSMENT FORM

Assessment Reference No.		**Area or activity assessed:**	
Assessment date			
Persons who may be affected by the activity (i.e. are at risk)	**Andrew Sotheran**		

SECTION 1: Identify Hazards - *Consider the activity or work area and identify if any of the hazards listed below are significant (tick the boxes that apply).*

1.	Fall of person (from work at height)		**6.**	Lighting levels		**11.**	Use of portable tools / equipment		**16.**	Vehicles / driving at work		**21.**	Hazardous fumes, chemicals, dust		**26.**	Occupational stress	
2.	Fall of objects		**7.**	Heating & ventilation		**12.**	Fixed machinery or lifting equipment		**17.**	Outdoor work / extreme weather		**22.**	Hazardous biological agent		**27.**	Violence to staff / verbal assault	
3.	Slips, Trips & Housekeeping	X	**8.**	Layout , storage, space, obstructions		**13.**	Pressure vessels		**18.**	Fieldtrips / field work		**23.**	Confined space / asphyxiation risk		**28.**	Work with animals	
4.	Manual handling operations		**9.**	Welfare facilities		**14.**	Noise or Vibration		**19.**	Radiation sources		**24.**	Condition of Buildings & glazing		**29.**	Lone working / work out of hours	
5.	Display screen equipment	X	**10.**	Electrical Equipment	X	**15.**	Fire hazards & flammable material		**20.**	Work with lasers		**25.**	Food preparation		**30.**	Other(s) - specify	X

SECTION 2: Risk Controls - *For each hazard identified in Section 1, complete Section 2.*

Hazard No.	Hazard Description	Existing controls to reduce risk	**Risk Level** (tick one)			Further action needed to reduce risks *(provide timescales and initials of person responsible)*
			High	Med	Low	
3	Tripping over wires	Cable management is at a minimum, none are currently properly cable managed and kept out of way			x	Sufficient cable management needed, cables tied together and moved out of way of feet
5	Eye strain from looking at a monitor	Current screen contrast and brightness is acceptable		x		To have periodic breaks from the screen
Name of Assessor(s)			**SIGNED**			
Review date						

Health and Safety Risk Assessments – continuation sheet

Assessment Reference No	
Continuation sheet number:	

SECTION 2 continued: Risk Controls

Hazard No.	Hazard Description	Existing controls to reduce risk	**Risk Level** (tick one)			Further action needed to reduce risks *(provide timescales and initials of person responsible for action)*
			High	Med	Low	

Name of Assessor(s)		**SIGNED**
Review date		

Appendix B - Log book

The log book for this project is a physical book and was handed to the School of Computer Science. Due to being a physical book, it cannot be inserted here.