Candidate Number: 3182E

Twitter to trade: The role of sentiment in cryptocurrency markets

In this dissertation I use Natural Language Processing to hunt for "animal spirits" in cryptocurrency markets. Namely, I study the relationship between market sentiment and prices of five major coins. I construct a new market sentiment measure by extracting opinion polarity from Twitter messages, using an original Deep Recurrent Neural Network I designed. My method outperforms the industry standard for sentiment classification. I then investigate the link between sentiment and price movements using a high-frequency event study. My results confirm that animal spirits are strong in cryptocurrency markets, that prices tend to determine sentiment, and hence bubbles are self-fulfilling.

Word Count: 7,494 words, including 16 lines of equations, 4.5 pages of tables and figures

1 Introduction

Cryptocurrency markets do not have fundamentals (Cheah and Fry, 2015), their trading actors are of low sophistication (Rauchs et al., 2018), and their prices are much more volatile than traditional assets (Dowd, 2014). Price movements tend to be large and happen fast, due to the changing market sentiment, anchored by social media *viz.* the crowd effect. In this dissertation, by market sentiment I mean the overall strength and orientation of opinion of market participants. Xu and Livshits (2018) show that cryptocurrencies are systematically exposed to "pump-and-dump" schemes; larger market players try to affect prices by hyping up the sentiment of small-time players using social media. Cheah and Fry (2015) suggest that bubble formation in Bitcoin could be due to self-fulfilling expectations. I hypothesise that, because of this, price movements should be related to overall market sentiment. I test that hypothesis in this dissertation by asking three questions:

1. Can Twitter data be used to construct a measure of market sentiment for cryptocurrency prices?
2. Is the market sentiment driven by cryptocurrency prices or vice-versa?
3. Do the impacts of market sentiment on prices vary and how so?

Answers to these questions should shed light on whether cryptocurrency markets are driven by fundamentals and/or speculation on sentiment. This, as well as how susceptible their prices are to manipulation. The answers therefore inform us both about the efficiency of markets and the government adoption/regulation of cryptocurrencies. Given that the Bank Of England is currently considering issuing a Central Bank Digital Cryptocurrency (Kumhof and Noone, 2018) (CBDC) such questions need to be answered in relation to its three principle functions as money (Dowd, 2014).

As the first step to answering the questions above I construct a measure of market sentiment using Twitter posts (tweets). To do that, I develop an original model for sentiment analysis, which has greater performance on unseen data than the commonly used Support Vector Machine (Ranco et al. (2015); Giaglis et al. (2015)). I use my model to classify the sentiment content of tweets. This analysis is then used to construct a new and unique measure of market sentiment, as well as an event detection procedure that builds on the past work of Ranco et al. (2015). My procedure combines new features to make it consistent with literature on economic narratives by J. Shiller (2017). It is able to detect 'sentiment events': periods of high-volume, high-polarity Twitter sentiment.

As such, this dissertation showcases the power of Natural Language Processing (NLP), when combined with econometrics, in answering economic questions. In addition, it contributes to the limited literature relating sentiment analysis of Twitter data to cryptocurrency markets. While McAteer (2014) and Giaglis et al. (2015) go some way towards opening up this field of study, their methodology employs techniques that I later find non-robust to residual autocorrelation. Hence I employ a robust, high-frequency event study observing the impact of sentiment events on cryptocurrency prices. I believe the results of this dissertation to be the first to robustly deal with these issues specifically for cryptocurrencies, especially on a high-frequency basis. However, my results can also be readily be applied to any form of economic sentiment analysis. For example, my method could be used to extract contextual information from the minutes of Central Bank governors meetings (Hansen et al., 2019). My methods therefore open up economics to the benefits of cutting-edge NLP.

My results suggest that prices cause sentiment and are strongly influenced by animal spirits: cryptocurrencies are therefore prone to self-fulfilling bubbles or market crashes. I observe that market sentiment acts as a propagation mechanism for second-round effects but its behaviour varies over cryptocurrencies and the polarity of sentiment events. This is consistent with both pump-and-dump schemes and trend-chasing behaviour (Xu and Livshits (2018); Greenwood and Nagel (2009)).

2 Data

To answer questions about the market conditions facing individual cryptocurrency traders, we need three overarching pieces of data: 1. Data containing opinions from an accessible source, that hosts a large number of market participants; 2. Data against which to estimate and test a sentiment analysis model; 3. Cyptocurrency pricing data with which we can model the interaction of sentiment and prices. The following data description is split according to this specification. Data are collected for the following coins: Bitcoin (BTC), Monero (XMR), Ethereum (ETH), Litecoin (LTC) and Ripple (XRP).

2.1 Streamed Twitter data

The data I use to calculate the market opinion of cryptocurrencies involves a 1% sample of all tweets for the 115 days 23 August to 15 December 2018. I live-streamed this sample, continuously, to my laptop using a combination of my own code and that from the python package `Tweepy`.[1] I filtered this sample such that tweets had to include phrases relating to a specific cryptocurrency. As an example, tweets about Bitcoin had to include 'bitcoin', 'btc',

[1] https://github.com/EconTriposIIBdissertation/dissertation/blob/master/Data%20Collection/Crypto_Tweets_2MySQL.py

'#bitcoin' or '#btc' in them. This filtering was not case-sensitive. Each tweet contained not only the text but also information on the time the tweet was published, the publisher's username, and the number of followers they had. Numbers of tweets collected for each cryptocurrency are given in Table 3.

2.2 Sentiment Analysis training data

The aim of most NLP, in particular sentiment classification, is to predict a target value (the probability of positive sentiment) using a textual input (the words of a tweet). Hence my sentiment prediction problem is one of supervised machine learning i.e. fitting a model on inputs and comparing its performance on labelled outputs. To train my sentiment prediction model I collected a sample of 99,988 generic tweets, with their `text` data and hand-labelled `sentiment`, from a competition on kaggle.com. The following table is an example of that data:

Sentiment	Text
1	Markets Update: Bitcoin Cash Leads the Pack With Double Digit Gains
0	Bitcoin #BTC will have high fees. The block size shouldn't be increased.
0	Let's be honest - everyone's tired of ICOs.
1	Ethereum (ETH) Price Showing Positive Signs Above 100 SMA
⋮	⋮

Sentiment = 1 if positive and 0 if negative.

Unlike the streamed twitter data, the Kaggle data had no unicode emojis, only smileys. It also had some other unusual features such as long strings of special characters e.g. '!?*-@'. Both the streamed and Kaggle data were therefore 'cleaned' with a custom text preprocessing algorithm I coded from scratch.[2] For example, in accordance with Agarwal et al. (2011), sequences of a repeated character were replaced by only three of that character e.g. 'cooooooool' was replaced with 'coool'. This preprocessing ensured that both sets of textual data were as similar as possible, for better generalisability of model performance/predictive ability.

2.3 Price data

The data on prices for cryptocurrencies was collected from the coindesk.com API. This is minutely pricing data for the 216 days between 5 May and 24 December 2018. Figure 9, in

[2] https://github.com/EconTriposIIBdissertation/dissertation/blob/master/Sentiment_Model/tweet_cleanerV2.py

the Appendix, displays the mean daily price of each cryptocurrency between these dates. For each coin the data is clearly non-stationary and heteroskedastic. They all also seem to follow the same trends - sharp price falls in early June, early August and mid-November with either recoveries or periods of maintenance in between.

3 Methodology

3.1 Sentiment Analysis

In order to construct a measure of market sentiment from Twitter, first a framework for quantifying sentiment must be created. This framework is derived from Computational Linguistics, in particular NLP. In sentiment analysis, polarity refers to the 'orientation' (Liu (2012), p. 18) of the sentiment contained in written or spoken language i.e. how positive, neutral or negative words, phrases or documents are. This polarity measure forms the basis for my Sentiment Measure. For convenience, when I refer to the sentiment of something this includes reference to its polarity.

To detect the sentiment of tweets, a Neural Network is used, which solves a supervised machine learning problem. Such techniques are used by Google (Wu et al., 2016) to perform text classification, translation, and generation. Goldberg (2018) demonstrates how Neural Networks take account of the *Distributional Hypothesis* of language (Harris, 1954) - hence giving Neural Networks theoretical underpinnings. He also shows their advantage in being able to drop the *markov-assumption*[3] of language and as such their ability to improve over Maximum Likelihood Estimators for the language modelling task. In addition to Goldberg noting their greater computational efficiency and perplexity scores[4], Le and Mikolov (2014), Socher et al. (2013), and Mariel et al. (2018) independently find large predictive improvements of Neural Networks over traditional approaches to language modelling.

I designed the following sentiment model architecture uniquely, using the intuition, technical knowledge, and recommendations contained within Goldberg (2018). In addition, I incorporated network features that have been found to perform well empirically in NLP literature.

[3]Future words only depend on a finite number of past words (Goldberg 2018, p. 106)
[4]Perplexity is a measure of success for language models discussed on page 106 of Goldberg (2018)

3.1.1 Neural Network Architecture

My Deep Neural Network was trained using the Kaggle dataset. The `text` data was used to predict the hand-labelled `sentiment` variable. Each tweet's `text` was encoded, passed through the Neural Network, and assigned a probability that it displayed positive sentiment. Positive (negative) sentiment should therefore have been assigned high (low) probabilities close to 1 (0). The predicted probability of positive sentiment was then compared against the actual `sentiment` of the text to give a loss score for each tweet. By adjusting the parameters of the Neural Network, the aggregate loss was minimised across all tweets to give the 'best' predictive model. The loss measure used for optimisation was binary cross-entropy, which is commonly applied to NLP classification tasks:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] \tag{I}$$

where N was the total number tweets in the training sample, y_i was the actual, labelled `sentiment` (0 or 1) for tweet i and $\hat{y}_i$ was the prediction given by the Neural Network for tweet i.

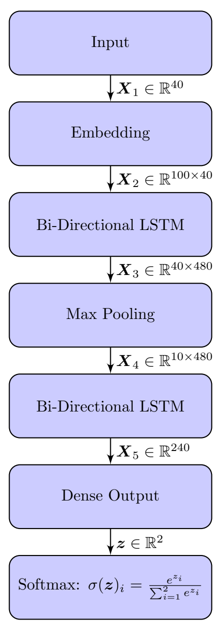

Figure 1: Sentiment Analysis Model Architecture

Figure 1 shows the layer architecture of my best performing neural network i.e. the model with the lowest cross validation loss score. The input layer encodes each tweet as 40 dimensional vector ($\boldsymbol{X}_1 \in \mathbb{R}^{40}$).[5] Each dimension refers to the next word in the sentence.[6]Each number in a dimension corresponds to one word in a dictionary based on the Kaggle dataset. The `Embedding` layer converts each word into a 100 dimensional vector ($\boldsymbol{X}_2 \in \mathbb{R}^{100}$). This 'embeddings' vector is the model's learned vector representation of a word. These vectors tend to have intuitive properties, such as if we take the vector for the word 'king' and subtract that for 'man' and add 'woman', the closest vector turns out to be 'queen'. Put another way, mathematical operations applied to the embeddings vectors tend to make semantic sense.

These embeddings vectors were trained using semi-supervised learning. Google has a set of 3 million word embeddings trained using the Word2Vec algorithm[7]. Words that existed in my model's dictionary, that had corresponding Google representations, were initialised to their Google values, otherwise using the Glorot Normal initialisation (He et al., 2015). Embeddings vectors were regularised with the square of the distance from their initialised values. This was to maintain the validity of mathematical operations over word vectors instilled by the initial google word representations e.g.

$$\mathbf{king} - \mathbf{man} + \mathbf{woman} = \mathbf{queen} \tag{II}$$

In turn it should have helped the model to generalise and avoid overfitting.

Embeddings are then fed into a type of a recurrent neural network (RNN) structure called a Bi-directional Long Short Term Memory (Bi-LSTM) layer. The Bi-LSTM component has been shown to consistently outperform other models for sentiment classification tasks by Li et al. (2015), Yin et al. (2017), and Tang et al. (2015). The Long Short Term Memory (LSTM) part is a form of RNN that allows the optimal prediction to use both the short and long-term context of words contained within a sentence. This represents an improvement over simple RNNs which tend to only be able to make use of short term context. For a full mathematical and intuitive understanding of LSTMs I would recommend Christopher Olah's blog.[8]

The outputs from this Bi-LSTM layer are then 'pooled'. Since each unit of the Bi-LSTM outputs a vector, 'pooling' involves combining 2 or more of these vectors in a way to reduce the number of inputs passed to the next layer. The model presented here pools groups of 4 sequential vectors by taking the maximum value for each dimension out of the 4 vectors. Johnson and Zhang (2014) found this sequential 'max pooling' effective for sentiment classification.

[5] 40 was the maximum number of words found in a Kaggle tweet
[6] Where sentences were shorter than 40 words the empty dimensions were padded with zeros.
[7] https://code.google.com/archive/p/word2vec/
[8] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

These pooled outputs are then passed into another Bi-LSTM layer because Li et al. (2015) show that this 'deep' structure to the network is effective for sentiment classification tasks. This adds greater non-linearity to the word-context transformations, enabling neural networks to find useful features, endogenously, without explicitly providing them e.g. Bi-grams, as inputs (Goldberg, 2018).

In turn the outputs from the second Bi-LSTM layer are optimally linearly combined, through a 'Dense' perceptron layer. They are then normalised, using the softmax function, to give a value for the probability that the text has positive sentiment. Dropout is used in the embeddings and both of the Bi-LSTM layers to provide further regularisation and allow the model to generalise better to unseen data. E. Dahl et al. (2013) found this form of regularisation effective for NLP tasks.

The model was trained on the 99,988 tweets from the Kaggle dataset, using 5 fold cross-validation. Its final performance was evaluated on a 20% hold-out validation set. After tuning hyper-parameters according to hold-out accuracy, the best model was then used to predict the sentiment of the unseen, livestreamed cryptocurrency tweets.

3.2 Vector Error Correction Model

Using the livestreamed tweets, a Vector Error Correction Model (VECM) was estimated for each cryptocurrency between the predicted `Sentiment` and `Price` variables. `Sentiment` refers to the Sentiment Measure I develop later in Equation IV. Model estimation code was written from scratch using the methods described in Lutkepöhl (2005) and some useful snippets from `statsmodels`.[9] This code gave the same estimates as those obtained from the Python library `statsmodels` except, unlike `statsmodels`, the code could now effectively handle missing data. Each model was fitted using the Engle and Granger (1987) Least Squares approach and the optimal lag order selected using the Akaike Information Criterion[10]. Results of the daily frequency VECM for Bitcoin are displayed in Figure 7. Cointegration was found at the 1% significance level, using Johansen's Max-Eigen test. Orthogonalised impulse response functions (IRFs), at the daily frequency, were created using `statsmodels` and are shown in Figure 8.

Shocks to `Sentiment` display only marginal persistence, while their effect on price is much more significant. This is consistent with conventional wisdom that sentiment triggers short-term, impulse buying decisions. These impulses quickly die off but leave behind a permanent, positive asset price impacts. The observed hysteresis of `Price` on `Price` and `Sentiment` on

[9] https://github.com/EconTriposIIBdissertation/dissertation/blob/master/VECM/my_VECM.py

[10] Lutkepöhl (2005) shows that the Schwarz Criterion is biased here

`Price` shocks could be due to market agents, with imperfect information, not being able to distinguish, based on their beliefs, whether price movements were due to fundamentals or speculative impulse buys - this forms a game-theoretic extension to my dissertation. As expected, shocks to `Price` impacted `Sentiment` but these impacts, although small, were also significantly permanent. That none of the shocks converged towards zero shows the non-stationarity of the system (Lutkepöhl, 2005). Given that all shocks/cross-shocks are positive and permanent, Bitcoin is therefore highly susceptible to self-fulfilling bubbles/crashes.

However, these results cannot be relied upon too much - there was significant evidence of residual autocorrelation using both the 20-minutely and daily frequency data. The sample residual autocorrelation plots for Bitcoin are given in Figure 6 in the Appendix. As such, there is strong evidence of some underlying structure that the VECM is not capturing, so we should be cautious of the conclusions drawn here. The residual autocorrelation is likely due to the problems of non-linearity in mean and variance, well-documented in quantitative finance (Cont (2001); Mandlebrot (1971); Sewell (2011)). Hence the main results of this dissertation follow an event study analysis, robust to these issues, outlined by Campbell, Lo and MacKinlay (1997).

3.3 Event Analysis

The following 5 subsections outline the event study methodology I employ, developed by Campbell et al. (1997).

3.4 Event Definition

Recall that Question 2 of this dissertation is whether market sentiment is driven by cryptocurrency prices or vice-versa. As such I focus on identifying "sentiment events" - periods of high-volume, high-polarity Twitter sentiment - to study their price impact. Key to this identification is a robust framework for event detection. Ranco et al. (2015) study the impact of Twitter events on stock prices using only tweet volume for event identification. To identify events they use a measure of the relative difference in Twitter activity, from a baseline, at a central point in time. They use a sliding window of $2L + 1$ days where L is the length of the event and define the measure $\phi(d_0)$ such that:

$$\phi(d_0) = \frac{TW_d - TW_b}{\max(TW_b, n_{min})} \tag{III}$$

where d_0 is the $(L + 1)^{th}$ time period in the sliding window, i.e. the central point, TW_d is the number of tweets in this period d_0, TW_b is the baseline or median number of tweets in the window and n_{min} is a minimum level of activity, 10 here, to avoid spurious event detection. A threshold value for $\phi(d_0)$ is then selected manually, to maximise detected events but minimise overlaps. All points in time where $\phi(d_0)$ is above this threshold, are then labelled as events.

There are three drawbacks to their method. First, detection is to some extent manual, requiring a person to make the threshold judgement decision, such that instantaneous detection is not possible. Second, detection is only based on tweet volume and takes no account of how Twitter can have market effects, based on the strength and polarity of opinion in an event. Finally, tweet volumes alone do not fully take account of how likely an event narrative is to spread to market participants. Events may be falsely detected where unimportant Twitter accounts spam tweets about cryptocurrencies. This is especially problematic for a high-frequency analysis, hence a more targeted measure is employed.

Robert Shiller, in his 2018 Marshall Lecture,[11] spoke about how narratives develop in ways similar to contagious diseases. Shiller gave examples of how narratives follow the dynamics of SIR models (Kermack et al., 1927) remarkably well: initially a few people are 'infected' with a narrative before that number suddenly jumps up, peaks, and then tapers off. Given we are interested in identifying an event by strong changes in narrative or, put another way, strong sentiment, there is motivation for updating the event identification methodology of Ranco et al. (2015) to reflect this. The major characteristics we want to incorporate are: (i) The event is marked by a sudden rise in the number of those 'infected' by a narrative. (ii) The total infected by an event does not necessarily tell us about the importance of that event e.g. everyone gets the common cold, not everyone gets Small Pox. (iii) Event dynamics are not only dependent on the number infected but also, to use the SIR terminology, the underlying infection, spread and recovery rates. (i) is consistent with the methodology of Ranco et al.. (ii) implies that, not only should we use some measure of the number of people tweeting, but also information about the strength and polarity of the sentiment in their narrative. (iii) additionally suggests we want to incorporate information relating to how quickly a narrative may spread. To deal with (ii), instead of simply counting the number of tweets made in a period, I incorporate a measure of sentiment too. I define the Sentiment Measure SM_t, in period t as

$$SM_t = \sum_{j \in \Omega_t} (Pr(\text{Sentiment}_j = 1) - 0.5) \times 2 \tag{IV}$$

where Ω_t is the set of indices for all tweets in time period t, Sentiment_j is the sentiment (1 if positive, 0 if negative) of the j^{th} tweet in period t. Hence the Sentiment Measure normalises a tweet that we know displays negative sentiment $[Pr(\text{Sentiment}_j = 1) = 0]$ to -1 and one that we know displays positive sentiment $[Pr(\text{Sentiment}_j = 1) = 1]$ to +1. Using

[11] He also wrote a discussion paper on the topic (J. Shiller, 2017)

this formula we incorporate both a measure of activity (number of tweets) and strength of sentiment (probability of sentiment being positive). The probability of the sentiment is used, instead of the {0,1} classification, due to inaccuracies in the sentiment classifier. It means we can make uncertain predictions [$Pr(\text{Sentiment}_j = 1) \approx 0.5$] meaningless as these tweets will add 0 to the Sentiment Measure. Also, the use of probability allows varying strengths of sentiment to be differentiated e.g. '...Amazing...' from '...Good...' as sentences with 'amazing'[12] should be given higher probabilities of displaying positive sentiment than those with 'good'. To account somewhat for (iii), all those tweets made by users with less than 1000 followers are removed from the set that contribute towards the Sentiment Measure. Here we are making an assumption that those users, with few followers, are unlikely to have narratives that spread and have significant impacts on market participants.

Building on Ranco et al. (2015), a new series is then constructed from the Sentiment measure. γ_t is defined as

$$\gamma_t = \frac{SM_t - SM_b}{\max(SM_b, L+1)} \tag{V}$$

SM_t is the Sentiment Measure at time t as defined previously. SM_b is a baseline of the Sentiment Measure. In this case it is the median value of the Sentiment Measure for the $L+1$ days, including and previous to, period t, where L is the event length. The term $L+1$ in the denominator is to avoid spurious detection of events. The baseline for an event is calculated relative to the L previous days because, according to SIR dynamics, a narrative should be recognised by a sudden initial jump in the Sentiment Measure rather than a peak at some point in the middle of a narrative's lifetime.

Finally, a time period t is defined as an event if its value of γ_t lay in the top 0.5% of all γ_t for the underlying period. This threshold was chosen using the same reasoning as statistical confidence intervals - a 0.5% chance of a time period being selected as a peak means those selected are highly likely to be due to significant narratives, rather than just statistical noise. Results and conclusions appeared robust to when a 1% threshold was used instead. A plot of the Sentiment Measure and identified events for Bitcoin is given in Figure 3.

3.5 Selection Criteria

Where identified events were within L days of each other, only the first of the events was chosen to be in the sample. Events before which there was missing data were were removed because this could have lead to anomalous results. No other restrictions were placed on event selection.

[12] Provided there is no other context to suggest negative sentiment

3.6 Normal and Abnormal Returns

As Campbell et al. (1997) show, the Market Model of returns tends to improve significantly over the Constant-Mean-Return model. Hence the expected return *viz.* normal returns, was estimated using the Market Model. This firstly involved creating a "market index" of cryptocurrencies, against which market returns could be calculated. The market index was created by taking the mean price of all 5 observed cryptocurrencies at every point in time. A linear regression of individual cryptocurrency returns on those of the market index was performed according to the specification

$$R_{it} = \beta_0 + \beta_1 R_{m,t} + \varepsilon_{it} \qquad R_{it} = \log(p_{it}) - log(p_{it-1}) \tag{VI}$$

where R_{it} is the return on cryptocurrency i at time t. $R_{m,t}$ is the market index return at time t. β_0, β_1 are parameters to be estimated.

3.7 Estimation Procedure

To prevent the identified events, subsequently analysed, from influencing parameter estimates, the regression was estimated on separate data prior to the event window. The event window, upon which the impact of sentiment events were analysed, ran for 115 days, from 23 August to 15 December 2018. The estimation window was the 108 days prior, from 6 May to 22 August.

3.8 Testing Procedure

With normal returns $\hat{R}_{it}$, for each time period t, estimated by OLS of the form:

$$\hat{R}_{it} = \hat{\beta}_0 + \hat{\beta}_1 R_{m,t} \tag{VII}$$

We can recover the residual "abnormal returns" $\hat{\varepsilon}_{it}$

$$\hat{\varepsilon}_{it} = R_{it} - \hat{R}_{it} \tag{VIII}$$

The abnormal returns $\hat{\varepsilon}_{it}$, are assumed to be jointly i.i.d. normal, with variance σ_i^2, under the null hypothesis that an event has no impact.

From this assumption we can derive statistics to test the alternative hypothesis that an event has an impact on returns. Defining the Cumulative Abnormal Return $\widehat{CAR_i(\tau_1}, \tau_2)$ for cryptocurrency i, between time periods τ_1 and τ_2, as

$$\widehat{CAR_i(\tau_1}, \tau_2) = \sum_{t=\tau_1}^{\tau_2} \hat{\varepsilon}_{it} \tag{IX}$$

Assuming that non-overlapping events are independent, then we can define the Average Cumulative Abnormal Return $\overline{CAR}_i$ as

$$\overline{CAR}_i = \frac{1}{N}\sum_{j=1}^{N} CAR_{ij}\widehat{(\tau_{j,1}}, \tau_{j,2}) \tag{X}$$

where j denotes the index of the j^{th} event that happens in the event window such that $\tau_{j,1}, \tau_{j,2}$ are the start and end time periods of event j.
Given, under jointly normal independent abnormal returns, that $Var(\hat{\varepsilon}_{it}) = \sigma_{it}^2 = \sigma_i^2$ then

$$Var(CA\widehat{R_i(\tau_1}, \tau_2)) = \sum_{t=\tau_1}^{\tau_2} \hat{\sigma}_i^2 \tag{XI}$$

Hence

$$Var(\overline{CAR}_i) = \frac{1}{N^2}\sum_{j=1}^{N}\sum_{t=\tau_{j,1}}^{\tau_{j,2}} \hat{\sigma}_i^2 \tag{XII}$$

As long as the number of time periods between $\tau_{j,1}$ and $\tau_{j,2}$ is the same, namely T, then due to the independence of abnormal returns assumption.

$$Var(\overline{CAR}_i) = \frac{T}{N}\sigma_i^2 \tag{XIII}$$

This enables us to create a test statistic for the significance of abnormal returns, in response to an event, that is consistent in the number of events recorded. The statistic

$$\overline{SCAR}_i = \frac{\overline{CAR}_i}{\sqrt{\frac{T}{N}\hat{\sigma}_i}} \overset{a}{\sim} N(0,1) \tag{XIV}$$

performs this role, where an estimator for $\hat{\sigma}$ is the sample variance of abnormal returns in the estimation window i.e

$$\hat{\sigma}_i^2 = \frac{1}{|\Omega_{est}| - 2}\sum_{\forall t_{est}\in\Omega_{est}} \hat{\varepsilon}_{it}^2 \tag{XV}$$

Where Ω_{est} is the set of all time periods t_{est}, in the estimation window. Testing using these statistics is carried out for each cryptocurrency, differentiating between sentiment events of positive and negative polarity.

4 Results

All results were generated in Python, using mainly the `Keras` API for sentiment modelling and `statsmodels` for the event study analysis. All code is my own, unless specifically stated, and is accessible on github at https://github.com/EconTriposIIBdissertation/dissertation.

4.1 Sentiment Analysis

Results for various sentiment models are shown in Table 1 and a graphical representation of performance (the ROC curve) is shown in Figure 2. In addition to my Deep Recurrent Neural Network (Deep RNN), used for the remainder of the analysis, results of two other Sentiment models are reported. Firstly an out-of-the-box sentiment classifier labelled 'Heuristic' in Table 1, from the Python package `NLTK`. This classifier works heuristically by picking up on key words e.g. 'brilliant', negations e.g. 'not', and boosters e.g. 'very', and adding up the sentiment scores of words, based on this information, to obtain a prediction.

The second classifier is a support vector machine (labelled 'SVM' in the Table 1). This classifier models text as a 'bag of words' (see Goldberg (2018) p. 93) and attempts to fit a linear hyperplane over sparse vector representations of sentences. It aims to separate the classes - positive and negative sentiment - by the largest 'margin'. For a full intuitive and mathematical explanation of SVMs see Ng (n.d.).

Ranco et al. (2015) choose this classifier for their analysis. Despite them reporting a very high accuracy, the results it gives seem untrustworthy.[13] The precision and recall scores are far below the classifier accuracy, which suggests inconsistencies in reported performance. Hence I compare its performance to alternative models.

	Deep RNN	SVM	Heuristic
Unseen Test Set Observations	24,997	29,997	76,401
Training Set Accuracy	81.3%	93.5%	-
Test Set Accuracy	78.9%	78.1%	71.2%
F1 score	79.0%	78.1%	70.2%
Precision/Recall (+)	72.8/82.4%	76.2/72.5%	74.0/51.9%
Precision/Recall (-)	84.8/76.1%	79.5/82.5%	70.0/86.1%

Table 1: Sentiment Model Classification Statistics

Table 1 shows that although the SVM model has the best training set accuracy of 93.5%, it is outperformed on unseen test data by the Deep RNN model i.e. the Deep RNN model has better ability to generalise information/avoid overfitting. Both classifiers greatly outperform the standard Heuristic model on both seen and unseen data. As such, this model is excluded from consideration but forms a useful baseline.

The Deep RNN's test accuracy of 78.9% is very close to the 79% of Li et al. (2015), on the dataset collated by Pang et al. (2002), using their own Bi-LSTM model. The consistency

[13] Ranco et al. (2015) p. 12

of my results with the NLP research standard supports the use of my Deep RNN. Figure 2 displays the Receiever Operating Characteristic (ROC) curves for each classifier. The ROC curve for my Deep RNN lies above that of the SVM. This implies that, for every level of classification certainty, the Deep RNN is better able to distinguish between positive and negative sentiment than the SVM.[14] Given its far better ability to generalise, to distinguish between sentiment polarities, and its higher accuracy, my Deep RNN is therefore chosen as the sentiment labelling model for the rest of this analysis.

Reciever Operating Characteristic Curves for Various Sentiment Models

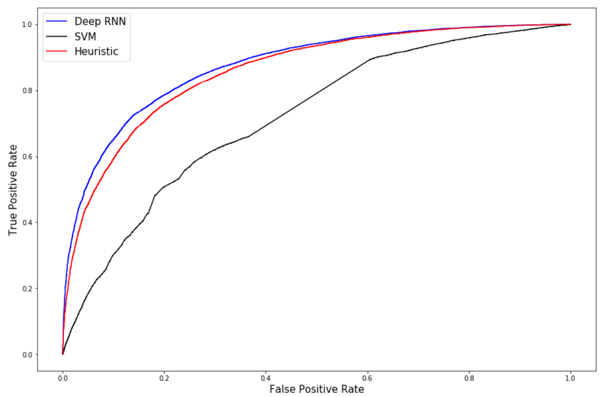

Figure 2: Reciever Operating Characteristic Curves for various sentiment models. ROC curves plot the true positive classification rate, of a model, against the false positive rate, for a given probability threshold of labelling something as True.

4.2 Event Study Analysis

4.2.1 Market Model

	BTC	ETH	XMR	LTC	XRP
Intercept	4e-05	-2e-05	2e-05	-3e-05	-2e-05
Slope	0.73*	1.15*	1.13*	1.02*	0.97*
R-Squared	0.78	0.67	0.61	0.76	0.7

Table 2: Market Model Regression Results. * - parameter significant at the 1% level

Table 4.2.1 displays the regression results for the Market Model of returns. All slope, but

[14]For a full explanation on interpreting ROC curves see Narkhede (2018)

no intercept, coefficients are significant at 1% and the high R^2 of each regression implies the Market Model significantly improves over the Constant-Mean-Return Model. The average R^2 from Table 4.2.1 is 0.704, which is equivalent to a 70.4% average reduction in the variance of abnormal returns, relative to the Constant-Mean-Return model (Campbell et al., 1997). This will improve the significance of an event's impacts given that lower abnormal return variance implies a higher signal-to-noise ratio around an event.

Although only a basket of 5 cryptocurrencies was used to construct the market price index, conclusions appear robust to the specification of the normal returns model - the Constant-Mean-Return Model generated very similar results when tested. Since the Market Model improves/reduces the variance of abnormal returns, I move forward using it for the estimation procedure.

4.2.2 Event Identification

Figure 3 shows the identified sentiment events and cumulative abnormal returns, for Bitcoin, over the period. Looking at the Sentiment Measure over time, we would expect detected events to be large spikes in this series above the baseline noise. This is almost exactly what we observe, with there being an diamond denoting an event at almost every large spike. Similar graphics for the other 4 coins are shown in Figure 5 in the Appendix.

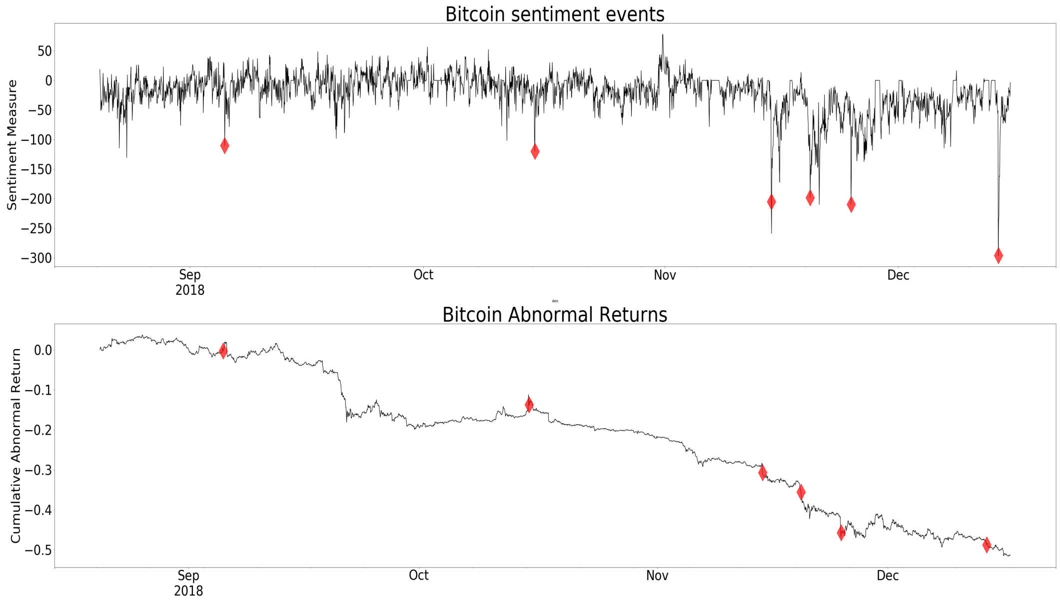

Figure 3: Identified hourly Bitcoin Twitter events and corresponding cumulative abnormal returns. Red diamonds denote dates at which negative sentiment events were detected.

Table 3 shows some key identification statistics for each cryptocurrency. For Bitcoin this tells us that 1. Over the 115 days from 23 August to 15 December 2018, 8,013,244 tweets are recorded as relating to Bitcoin. 2. Only 29.9% of these tweets are retained in the Sentiment Measure when filtered by whether the 'tweeter' had more than 1,000 followers. 3. When event identification is performed at the 20-minutely frequency, an event is identified by a spike in the Sentiment Measure such that it's value of

$$\gamma_{t,20min} = \frac{SM_t - SM_b}{\max(SM_b, L+1)} \tag{XVI}$$

is greater than 1.426. 4. Likewise, when event identification is performed at the hourly level the threshold value of $\gamma_{t,hourly}$ is 9.78.

Focusing particularly on the hourly event thresholds, values are highly variable, ranging from 1.27 to 9.78. This means what is considered an event varies massively by cryptocurrency. For Ethereum an event is identified if the Sentiment Measure is 494% above the baseline median. For Monero the equivalent value is 127%. This is important to notice as large variation in thresholds for events hints at potential limitations of my sample size: the sample may not have included many (or any) 'true' sentiment events, for certain cryptocurrencies. By 'true' sentiment events I mean those events characterised by a strong Twitter narrative, that have real implications for prices. In turn this would bias conclusions as non-events, with insignificant market dynamics, identified as true events, could be used to infer a lack of interaction between Twitter and crypto prices.

Since the 20-minutely event thresholds were suspiciously low relative to the hourly thresholds, and the identified Sentiment Measure spikes were far less clear than those at the hourly frequency, the 20-minutely results were dropped in favour of the hourly frequency study. Graphical results for the 20-minutely frequency are still given in the Appendix by Figure 10.

	Bitcoin	Litecoin	Ethereum	Monero	Ripple
Number of recorded tweets	8,013,244	930,394	3,319,875	226,270	2,012,360
Percentage of Tweets Retained	29.9%	32.2%	30.7%	26.61%	23.92%
20 Minutely event threshold $\phi_{M,20min}$	1.426	0.363	0.884	0.211	0.531
Hourly event threshold $\phi_{M,hour}$	9.78	1.88	4.94	1.27	3.12

Table 3: Tweets and event identfication statistics by cryptocurrency

4.2.3 The dynamics of sentiment & prices and direction of causality

Graphical results of hourly cumulative abnormal returns, in response to sentiment events, are shown in Figure 4. Where a positive (negative) event was identified, the resulting impact on price returns is traced out by a green (red) line. The 1% confidence intervals for significant, non-zero abnormal returns are shown by the corresponding coloured, dashed lines. Returns were normalised to zero at the time of the event, to clearly analyse the price dynamics around this.

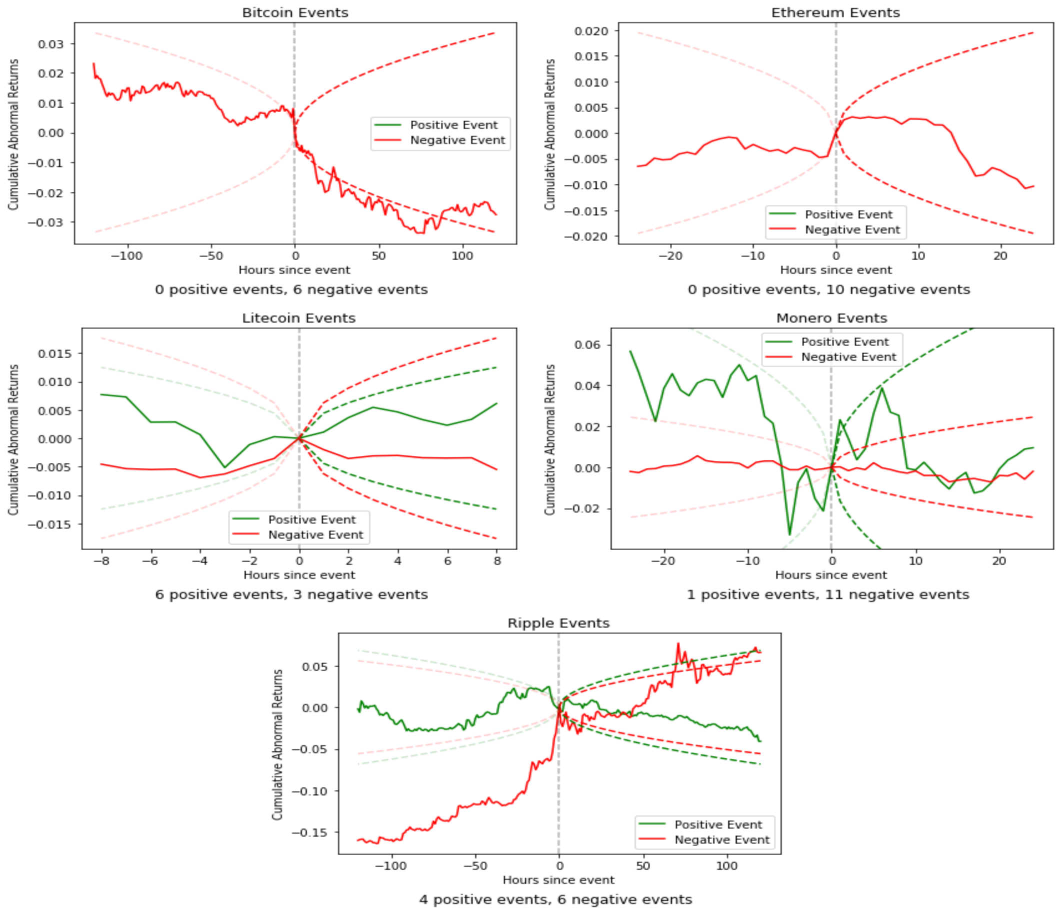

Figure 4: Average Hourly Cryptocurrency returns around an identified sentiment event for each of 5 cryptocurrencies. Cumulative abnormal returns are normalised to 0 at the time of the event and 1% confidence intervals are shown by the dashed lines.

For Monero and Litecoin the impact of a sentiment event appears to be mostly insignificant. This is seen by the fact that the cumulative abnormal returns (CAR) lie within the confidence intervals at pretty much all stages. Although Monero exhibits significant price changes either side a positive event, it is hard to draw inference as only one of these positive events was recorded. This is a limitation of my work - I only collected data over 115 days.[15] While an event study is robust to small samples, more observations and events would have increased the significance of my results and hence the strength of inference.

For Bitcoin, Ethereum and Ripple the CAR dynamics around an event are much more highly

[15] According to current Twitter pricing, the market value of my dataset is still up to $31,565.48

significant. Looking at Bitcoin, for which no positive events are identified, we see the CAR start a large decline about 10 periods (10 hours) before a negative event. The decline then continues until about 75 hours after the event before levelling off. This implies that an initial decline in the price of Bitcoin causes people to display large amounts of negative sentiment over Twitter. The post-event dynamics suggest that this sentiment is the mechanism for second-round effects - prices continue to decline much further and for much longer than the initial pre-event movement. Hence, for Bitcoin, an initial price decline causes Twitter panic leading to a self-fulfilling crash - the effects of which appear to be persistent.

For Ethereum again only negative events are detected and similar causality appears to exist - prices to sentiment. This time however price movements are in the opposite direction: In response to a significant price rise, within an hour a negative sentiment event is recorded. Prices then rise slightly further, but insignificantly, before levelling off. The price fully reverts to its pre-event level about 15 hours post-event. This behaviour is indicative of animal spirits: Price rises based on speculation cause less-experienced, lay traders to jump on the trend and overvalue the coin. Professional traders display negative sentiment over Twitter, since they believe/know the coin is overvalued, which halts prices after about an hour. Ultimately the price rises are reversed as this sentiment/knowledge feeds through the market. The market is eventually revalued to a (more) stable price.

Looking at the bottom graph of Figure 4, both positive and negative events are identified for Ripple. Like Ethereum, we observe price movements of the opposite sign to sentiment, in the lead up to an event. For negative (positive) events this means significant preceding price rises (falls). For positive events, similar dynamics are observed to that of Ethereum: A significant initial price fall, based on speculation, leads lay traders to jump on the trend and undervalue the coin. Professional traders meanwhile display a high volume of positive sentiment, over Twitter, concerning the coin's undervaluation. This sentiment feeds through the market to lay traders, halting price falls 2 hours post-event, before there is a substantial market correction of around 2%. The price then steadily revalues to a new level. Unlike Ethereum, in pretty much immediate response to a negative event, prices move significantly downwards. This behaviour is highly consistent with the pump-and-dumps observed by Xu and Livshits (2018): The 'pump' phase leads to significant price rises before participants cash out. This results in a 'dump' - immediate and significant price falls as traders cash out. In turn it triggers a negative sentiment event and further price falls, that we observe until about 20 hours post-event. Like Bitcoin, the price of Ripple is then sustained on an upwards trend. This suggests Ripple is also susceptible to self-fulfilling bubbles/panic and is driven by animal spirits.

There are a few key aspects to take away from this. (1) Cryptocurrency markets are heavily influenced/driven by animal spirits. (2) Prices cause sentiment, with sentiment acting as the propagation mechanism for second-round effects. (3) The dynamics of prices and sentiment are heterogeneous across markets: sentiment events can either counteract price movements (Ripple and Ethereum) or perpetuate them (Bitcoin). (4) The three largest cryptocurrencies by market capitalisation are the only ones with significant twitter/price interaction.

5 Conclusions

This dissertation used recommendations from the field of Natural Language Processing (NLP) to create an original and unique sentiment prediction algorithm. It then applied this algorithm to the particular case of Twitter, in order to study the interaction between market sentiment and cryptocurrency prices. Therefore, the main contribution of this work is my original sentiment prediction methodology. Especially its ready applications to wider economic theory. By applying cutting-edge NLP research to economic problems, vast areas of research are opened up. For example, the formation of inflation expectations, proxied by sentiment instead of surveys, in response to macroeconomic news (Angel Garcia and Werner, 2018). Not only this, but existing research can use updated and more powerful techniques e.g. the language models used for determining the response of long-run interest rates to Central Bank communication (Hansen et al., 2019).

When applied to the specific case of market sentiment relating to cryptocurrencies, my model was able to extract and quantify animal spirits from Twitter. It confirmed that, because cryptocurrencies have no fundamentals (Garcia et al., 2014), people's animal spirits drive the market. In particular, the idea that cryptocurrencies are prone to self-fulfilling bubbles (Cheah and Fry, 2015) was verified. These speculative tendencies have important consequences for the government adoption and regulation of cryptocurrencies (Dowd, 2014).

The event study analysis I employed forms a robust synthesis between a number of works. Ranco et al. (2015), Giaglis et al. (2015), and McAteer (2014) study the interaction of Twitter and financial market prices. However, my methodology combined their sentiment analysis, event study, and cryptocurrency aspects into one application. I found that (1) prices cause sentiment and never the other way around. (2) Sentiment acts as the propagation mechanism for second-round effects. (3) The dynamics of prices and sentiment are heterogeneous across markets: sentiment events can either counteract price movements or perpetuate them. (4) Only the largest cryptocurrencies by market capitalisation have significant twitter/price interaction.

My results highlight the power of NLP in only one application to economics. It should therefore be borne in mind that sentiment analysis goes beyond using Twitter to trade.

References

[1] Agarwal, A., Xie, B., Vovsha, I., Rambow, O. and Passonneau, R. (2011), 'Sentiment analysis of twitter data'.

[2] Angel Garcia, J. and Werner, S. (2018), 'Inflation news and euro area inflation expectations', IMF Working Papers **18**, 1.

[3] Campbell, J., Lo, A. and MacKinlay, C. (1997), The Econometrics of Financial Markets, Princeton University Press.

[4] Cheah, E.-T. and Fry, J. (2015), 'Speculative bubbles in bitcoin markets? an empirical investigation into the fundamental value of bitcoin', Economics Letters **130**, 32 – 36.
URL: *http://www.sciencedirect.com/science/article/pii/S0165176515000890*

[5] Cont, R. (2001), Empirical properties of asset returns: stylized facts and statistical issues, in 'Quantitative Finance Volume 1', Vol. 1 of 2, Institute of Physics Publishing, pp. 223–236.

[6] Dowd, K. (2014), 'New private monies: A bit-part player?', Institute of Economic Affairs Monographs .
URL: *https://iea.org.uk/wp-content/uploads/2016/07/New%20Private%20Monies%20-%20Kevin%20Dowd.pdf*

[7] E. Dahl, G., N. Sainath, T. and E. Hinton, G. (2013), Improving deep neural networks for lvcsr using rectified linear units and dropout, Vol. 26, pp. 8609–8613.

[8] Engle, R. F. and Granger, C. W. J. (1987), 'Co-integration and error correction: Representation, estimation, and testing', Econometrica **55**(2), 251–276.
URL: *http://www.jstor.org/stable/1913236*

[9] Garcia, D., Tessone, C., Mavrodiev, P. and Perony, N. (2014), 'The digital traces of bubbles: Feedback cycles between socio-economic signals in the bitcoin economy', Journal of the Royal Society, Interface / the Royal Society **11**.

[10] Giaglis, G., Georgoula, I., Pournarakis, D., Bilanakos, C. and Sotiropoulos, D. (2015), Using time-series and sentiment analysis to detect the determinants of bitcoin prices.

[11] Goldberg, Y. (2018), Neural Network Methods for Natural Language Processing, Morgan & Claypool.

[12] Greenwood, R. and Nagel, S. (2009), 'Inexperienced investors and bubbles', Journal of Financial Economics **93**(2), 239 – 258.
URL: *http://www.sciencedirect.com/science/article/pii/S0304405X09000646*

[13] Hansen, S., McMahon, M. and Tong, M. (2019), 'The long-run information effect of central bank communication', SSRN Electronic Journal .

[14] Harris, Z. (1954), 'Distributional structure', WORD pp. 146–162.

[15] He, K., Zhang, X., Ren, S. and Sun, J. (2015), Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in '2015 IEEE International Conference on Computer Vision (ICCV)', pp. 1026–1034.

[16] J. Shiller, R. (2017), 'Narrative economics', American Economic Review **107**, 967–1004.

[17] Johnson, R. and Zhang, T. (2014), 'Effective use of word order for text categorization with convolutional neural networks', CoRR **abs/1412.1058**.
URL: *http://arxiv.org/abs/1412.1058*

[18] Kermack, W. O., McKendrick, A. G. and Walker, G. T. (1927), 'A contribution to the mathematical theory of epidemics', Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character **115**(772), 700–721.

[19] Kumhof, M. and Noone, C. (2018), Staff working paper no. 725: Central bank digital currencies design principles and balance sheet implications, Technical report, Bank of England.

[20] Le, Q. V. and Mikolov, T. (2014), 'Distributed representations of sentences and documents', CoRR **abs/1405.4053**.
URL: *http://arxiv.org/abs/1405.4053*

[21] Li, J., Jurafsky, D. and Hovy, E. H. (2015), 'When are tree structures necessary for deep learning of representations?', CoRR **abs/1503.00185**.
URL: *http://arxiv.org/abs/1503.00185*

[22] Liu, B. (2012), Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers.

[23] Lutkepöhl, H. (2005), New Introduction to Multiple Time Series Analysis, Springer.

[24] Mandlebrot, B. (1971), When can price be arbitraged efficiently? a limit to the validity of the random walk and martingale models, in O. Eckstein, ed., 'The Review of Economics and Statistics, Vol. 53, No. 3', The MIT Press, pp. 225–236.

[25] Mariel, W. C. F., Mariyah, S. and Pramana, S. (2018), 'Sentiment analysis: a comparison of deep learning neural network algorithm with SVM and naïve bayes for indonesian text', Journal of Physics: Conference Series **971**, 012049.
URL: *https://doi.org/10.1088%2F1742-6596%2F971%2F1%2F012049*

[26] McAteer, C. (2014), 'Twitter sentiment analysis to predict bitcoin exchange rate'.
URL: *https://scss.tcd.ie/publications/theses/diss/2014/TCD-SCSS-DISSERTATION-2014-055.pdf*

[27] Narkhede, S. (2018), 'Understanding auc - roc curve'.
URL: *https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5*

[28] Ng, A. (n.d.), 'Cs229 lecture notes. part v. support vector machines'.
URL: *http://cs229.stanford.edu/notes/cs229-notes3.pdf*

[29] Olah, C. (2015), 'Understanding lstm networks'.
URL: *http://colah.github.io/posts/2015-08-Understanding-LSTMs/*

[30] Pang, B., Lee, L. and Vaithyanathan, S. (2002), 'Thumbs up? sentiment classification using machine learning techniques', EMNLP **10**.

[31] Ranco, G., Aleksovski, D., Caldarelli, G., Guido, G. and Mozetic, I. (2015), 'The effects of twitter sentiment on stock price returns'.

[32] Rauchs, M., Blandin, A., Klein, K., Pieters, G., Recanatini, M. and Zhang, B. (2018), 2nd global cryptoasset benchmarking study, Technical report, University of Cambridge Judge Business School.

[33] Sewell, M. (2011), 'Characterization of financial time series', UCL Department of Computer Science Research Notes .

[34] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A. and Potts, C. (2013), 'Recursive deep models for semantic compositionality over a sentiment treebank', EMNLP **1631**, 1631–1642.

[35] Tang, D., Qin, B., Feng, X. and Liu, T. (2015), 'Effective lstms for target-dependent sentiment classification', CoRR **abs/1512.01100**.
URL: *http://arxiv.org/abs/1512.01100*

[36] Unknown (2017), 'Twitter sentiment analysis, kaggle inclass competition'.
URL: *https://www.kaggle.com/c/twitter-sentiment-analysis2/data*

[37] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., ukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M. and Dean, J. (2016), 'Google's neural machine translation system: Bridging the gap between human and machine translation'.

[38] Xu, J. and Livshits, B. (2018), 'The anatomy of a cryptocurrency pump-and-dump scheme'.

[39] Yin, W., Kann, K., Yu, M. and Schütze, H. (2017), 'Comparative study of CNN and RNN for natural language processing', CoRR **abs/1702.01923**.
URL: *http://arxiv.org/abs/1702.01923*

6 Appendix

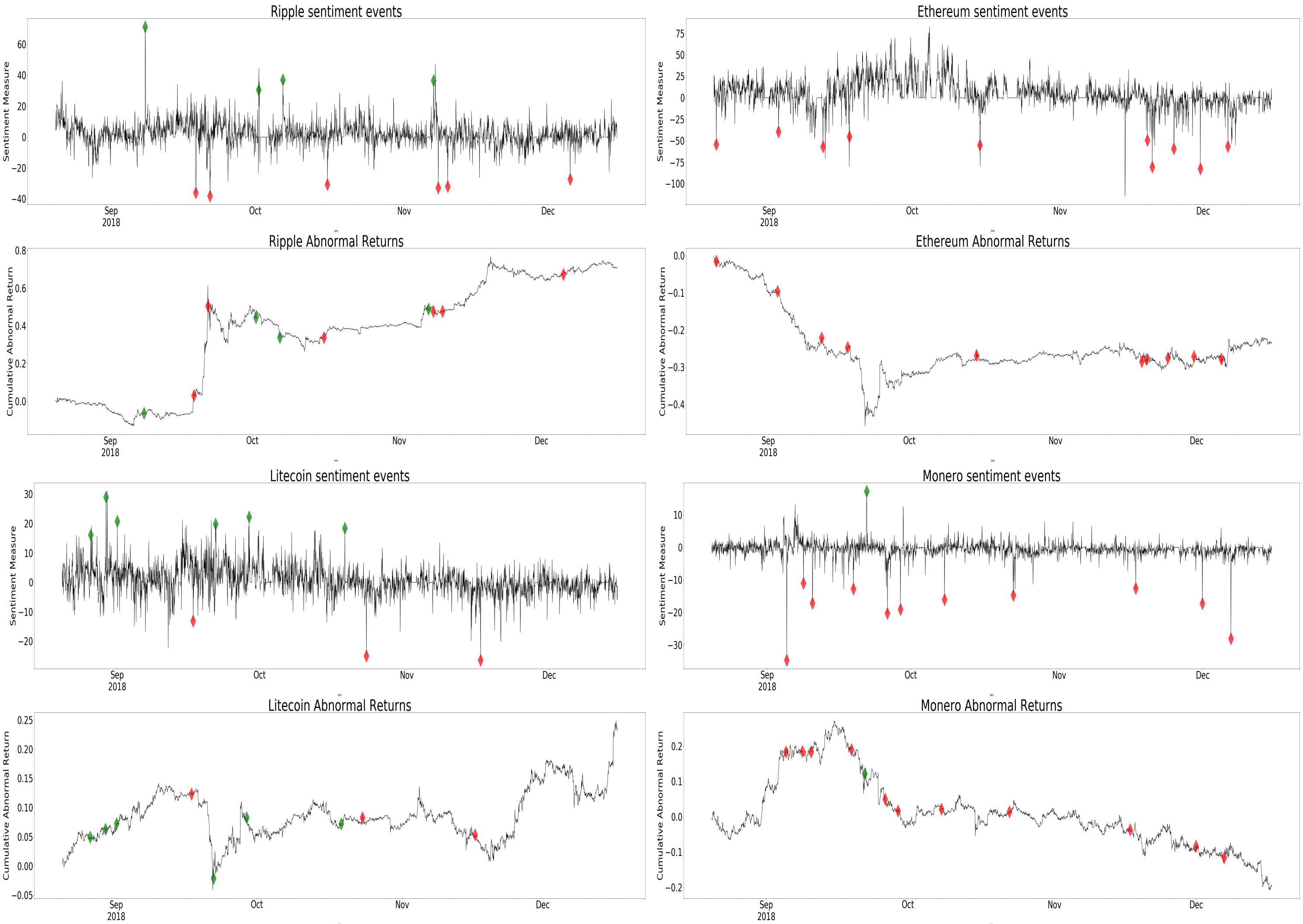

Figure 5: Identified hourly Twitter events and corresponding cumulative abnormal returns. Red diamonds denote identified negative sentiment events, green denote positive sentiment events.

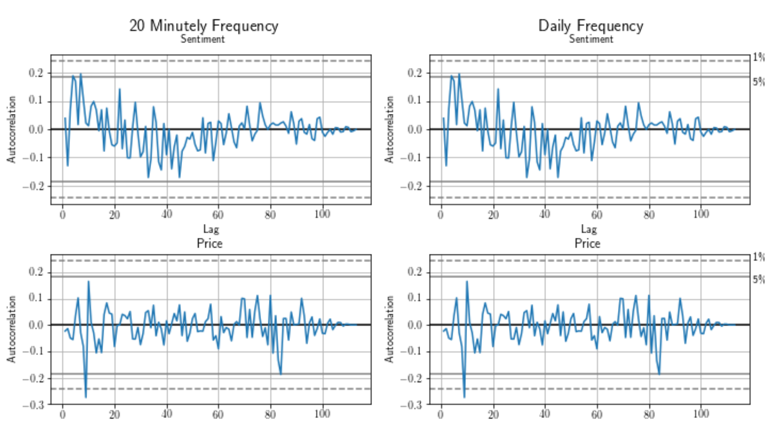

Figure 6: Bitcoin VECM residual autocorrelation plots. A 5% confidence interval is given by the solid gray line. A 1% confidence interval is given by the dashed grey line.

$$\Delta y_t = \begin{bmatrix} -0.5074^* \\ 1586.5 \end{bmatrix} \begin{bmatrix} 1.000^* & -0.000015^* & -0.3868^* \end{bmatrix} y^+_{t-1} + \begin{bmatrix} 0.1457 & 0.000001 \\ 2004.9 & 0.1352 \end{bmatrix} \Delta y_{t-1} + \epsilon_t$$

$$y_t = \begin{bmatrix} Sentiment_t \\ Price_t \end{bmatrix} \qquad y^+_t = \begin{bmatrix} Sentiment_t \\ Price_t \\ 1 \end{bmatrix}$$

*Coefficient significant at the 5% level.

Johansen's cointegration test statistic for 1 cointegrating vector: 32.39 (significant at 1% level.)

Figure 7: Matrix representation of Bitcoin VECM

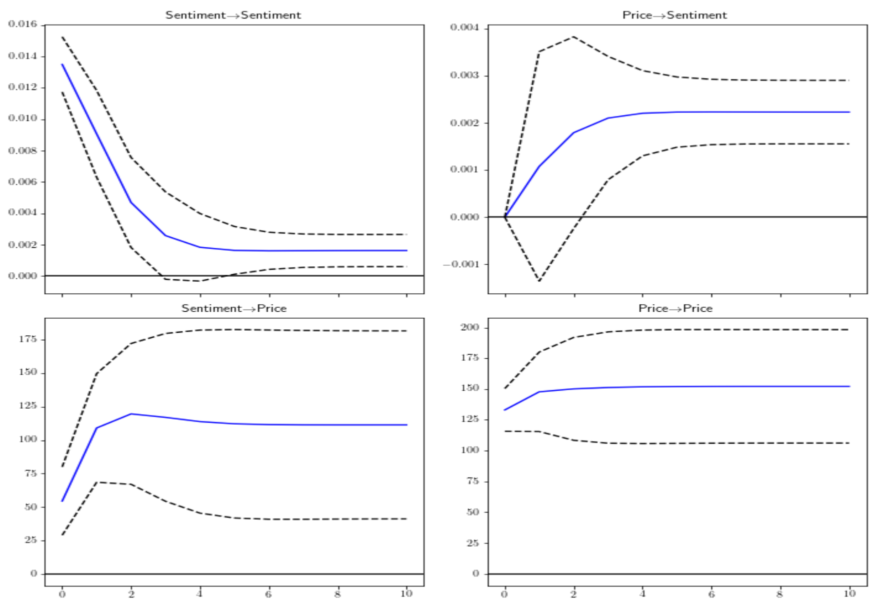

Figure 8: Orthogonalised impulse response functions for Bitcoin at the daily frequency. Blue lines show expected paths of variables in response to an orthogonalised 1 standard deviation impulse. 5% confidence intervals are shown by the dashed black lines.

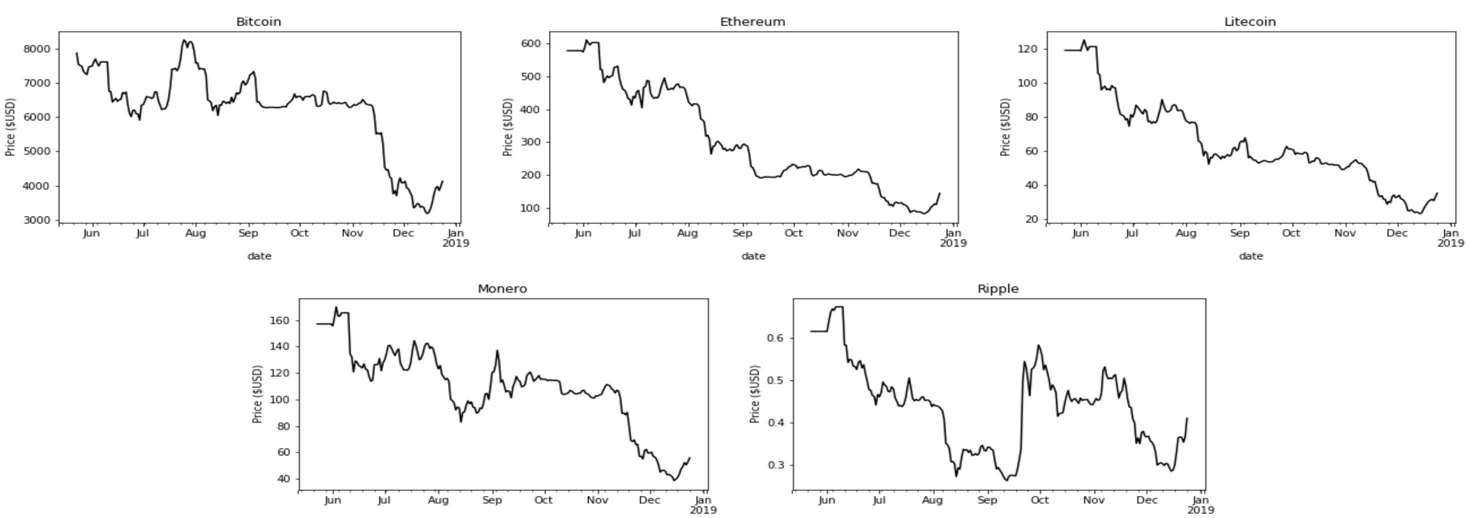

Figure 9: Prices of 5 major cryptocurrencies over time

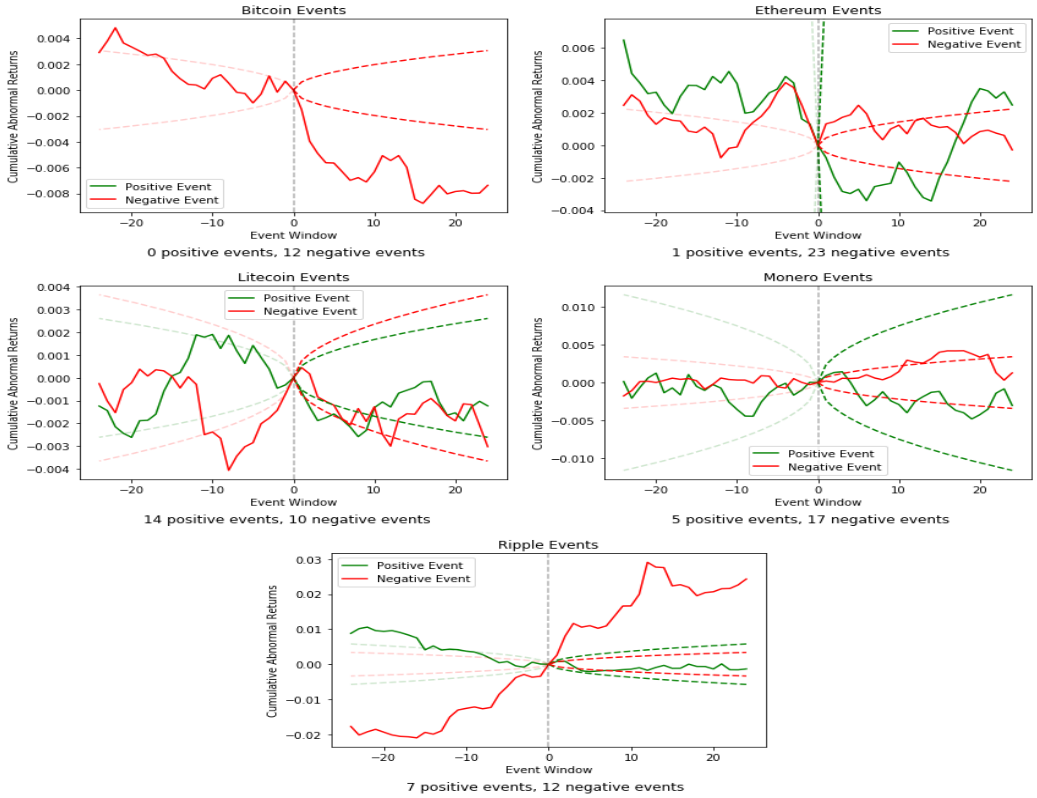

Figure 10: 20 Minutely Frequency Cryptocurrency events. The average cumulative abnormal return around an identified sentiment event for each of 5 cryptocurrencies. Cumulative abnormal returns are normalised to 0 at the time of the event and the 1% confidence intervals are shown by the dashed lines.