Predicting short term log returns for Ethereum using transactional network graph characteristics

Vadim Sokolov*

April 25, 2018

Abstract

Over the past two years, the blockchain technology has witnessed a growth in user interest.Ethereum is a blockchain based open source software platform that allows decentralized applications to be built on it. Ethereum implements its own version of a blockchain and conforms with the growth in user interest across major cryptocurrencies. Like most cryptocurrencies, Ethereum maintains a publicly available ledger, which allows for the network of transactional interactions to be analyzed. In this project, summary graph features and additional subgraph characteristics of the Ethereum network used to predict log returns. We assess the effects of adding multiple types of subgraph features on predicting log returns. We assess the performance of traditional machine learning models and neural network models on predicting the log return for this time series data.

Keywords: Ethereum, Subgraph, Network, Price Prediction ,Deep Learning

*Vadim Sokolov is Assistant professor at George Mason University. email:vsokolov@gmu.edu

1 Introduction

In simple terms, a blockchain is defined as a distributed and immutable public ledger of transactions that are verified by a majority consensus of system participants [15]. Ethereum uses the core concepts of a blockchain, but differs in the fact that it is a distributed public blockchain network. A key differentiating feature is the Ethereum Virtual Machine (EVM), a Turing complete software used to build and run decentralized applications (Dapps) on the Ethereum network.

The existence of Dapps, general interest in cryptocurrencies and launching initial coin offerings led to some record breaking numbers for Ethereum – it beat Bitcoin's transaction volume by over 200 % in Q4 2017 and the network hashrate increased by over 40 % in the same time [14]. This has also led to a growth in the number of Dapps, with at least 1200 curated applications available [1]. During this time, the price of Ethereum has also fluctuated, reaching a high of almost $1426.86 ETH. Ethereum strongly encourages the development of Dapps by making it relatively easy to write contract account code in Ethereum's solidity programming language [16], which also allows for the possibility of algorithmic trading.

Most stock market price prediction methods consider a mix of historical data, market sentiment, crowd sourced knowledge repositories, macro and micro economic factors. Cryptocurrencies allow transactional network graphs to be generated, which allows additional features to be added to the dataset to predict future returns. The core idea here is that frequency of occurrence of a subgraph and its features has a relationship with the direction in which the fluctuation proceeds.

We believe that considering the subgraph characteristics of the Ethereum network can be a good indicator of price fluctuations. We aim to consider a list of summary subgraph features and capture information that pertains to describing the network motifs of each subgraph.

2 Previous Work

The role of subgraph characteristics being used to define network features has been discussed in a wide variety of applications, ranging from biological networks [36], social networks [17], transportation [43] and computational social networks [38]. The idea of networks subgraphs being used in transaction records has often been limited by the fact the transaction records are often not publicly available, unless researchers are a part of the ecosystem [26]. Since cryptocurrencies allow for an open transaction ledger, this allows for network features to be extracted from them.

25%	50%	75%
8.221	11.946	220.075

Table 1: Descriptive statistics for Close Price of Ethereum

Bitcoin has accounted for a large fraction of research into this area, ever since the launch of Satoshi Nakamoto's Bitcoin whitepaper [37]. There has been research that focuses on tracking illegal activity [19], money laundering [18], network centralities [5] and power distributions [33]. On the other hand, there is relatively less research on Ethereum specifically, but it does appear in macro-level research that focuses on security [34], proof of work [29], economics [11] and statistical analysis [9].

An area of research that has gained popularity over time is price prediction. As with other aspects of cryptocurrency research, the focus has been heavily skewed in favor of bitcoin. Many different approaches have been considered, including quantitative analysis [39], trading patterns [23] and user sentiment [30] [41]. There has also been some research that focuses on using network features to predict price [22] [32], both in general and specific to Ethereum [25]. There have been different time deltas considered to predict the price of Bitcoin. While some consider time deltas for a much longer slot, we restrict our work to time slots in hours or lesser. Hegazy and Mumford consider a time delta of 8 minutes to compute an exponentially smoothed bitcoin price and feed its first five left derivatives as features into a decision tree, with an

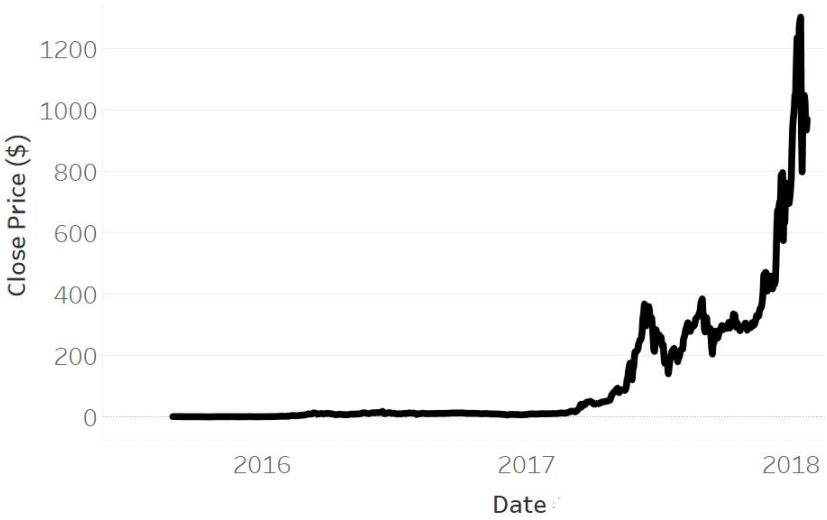

Figure 1: Close Price of Ethereum in USD over time

accuracy of 57.11 % [25]. Madan, Saluja and Zhao consider a time delta of 10 minutes with the price of bitcoin as a feature, achieving an accuracy of 57.4 % [35]. Chen, Narwal and Schultz consider a time delta of 1 hour and consider previous price points to feed into an ARIMA model, achieving an accuracy of 61.17 % [10].

3 Dataset

There are two primary datasets under consideration. The first dataset is the close price of Ethereum sampled in one hour intervals between 31 August 2015 and 24 Jan 2018. The data has been gathered from Coindesk [13].Coindesk is a leading digital media and information services provider for the crypto asset and blockchain technology community. This dataset is plotted in Figure 1 and summary statistics for the same are mentioned in Table 1. The second dataset is the Ethereum transaction database, which has been gathered by using the Ethereum web3.js API [2]. The API is used to communicate with a local node through remote procedure calls, enabling users to query information about transactions contained in a block. After the raw information is gathered, data is merged and data fields which are not useful are eliminated. The data is converted from hexadecimal to human

readable format, after which the information is used to extract network features. Summary details about the dataset are mentioned in Table 2. Data dictionary for this information is mentioned in table 3.

Mean transactions/hour	6866.901
Max transactions/hour	77646

Table 2: Summary details for Ethereum transaction data

From	Address from which Ether is being sent
Gas	Gas charged for transaction
GasPrice	Gas price for transaction
Input	Contains hex code if transaction is a contract, else is zero
To	Address to which Ether is being sent
Value	Value of Ether
GasUsed	Gas used for transaction
Miner	Block Miner
Number	Serial number of transaction
Timestamp	Unix timestamp

Table 3: Data description of Ethereum transaction data

There are three distinct types of network motifs that are observed in the Ethereum graphs generated for various time deltas. The first is networks that form a hub like structure , with connections existing bidirectionally from the central node. While there is some assortativity that is noticed between the central nodes of each hub , the number of such connections that exist is marginal when compared to other types of connections that are observed. Upon heuristic examination, it is seen that these kind of structures are usually sue to cryptocurrency exchanges that serve as a center for a large number of transactions to occur through them. The second kind of network motif that is seen is a cascading structure. The number of nodes that exist in these

cascading structures is between 3 and 8, based on a heuristic assessment. The third kind of network motif is a pair-wise structure, which points to a transaction occurring between two individual wallets. There are also some clique like network motifs that are seen, but their count is negligible when compared to the three motifs mentioned.

3.1 Data pre-processing

As can be seen from the Ethereum close price graph in Figure 1, there is not much movement in price for the first year that Ethereum was active. This led us to believe that there was a sizeable variance in the dataset. We reduced this by removing the first 7000 rows (∼290 days) of the data. The choice to do so was arbitrary, with the intention being to not give too much credence to the features that exist during the early days of Ethereum. The same experiments were performed on both the truncated and untruncated data, with no significant difference in outputs. To ensure consistency, the results for the dataset after data truncation are mentioned in the paper.

The dataset was split in a 0.8/0.1/0/1 ratio for training/validation/ test. The split is ordered in time, with the latest timestamp representing the validation set. Hyperparameters are tested by feeding the training set for evaluation. Final results are obtained and evaluated based on metrics for training and test sets.

Using the price data, the log return is generated. We decided not to use features traditionally used in market price prediction like index prices, ROIC and Williams Percentage range [6]. The reason for this is to give precedence to network features. All the transactions for a time delta are gathered from the Ethereum transaction data and represented as a transactional graph. The summary graph features are then extracted using the network library [24]. The list of features is mentioned in Table 4.

Edge count	Count	Node count
Connected components count	Transitivity	Centrality
Average clustering	Degree	Density

Table 4: Summary graph features

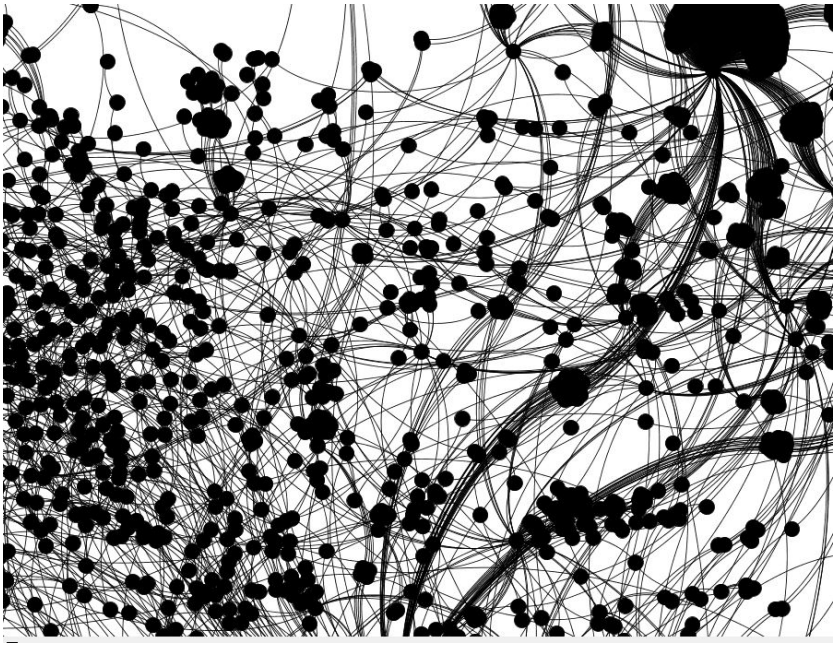

Figure 2: A network graph of Ethereum transactions for a time delta

After this is done, additional graph features are extracted. This is done in the following manner:

1: For transactions in a time delta, graph G is generated with nodes and edges n and e.

2: For each n in G, the edge count b is obtained.

3: The counts of each unique edge count b are obtained as f and added as features.

4: To ensure data consistency across time deltas, different f values are normalized in length by adding zeroes.

5: The data is normalized by using a minimum-maximum scaler with range (0,1).

6: The process is repeated for all time deltas

4 Deep Learning for Ethereum

Multiple models were assessed on the task of predicting the log return of the price of Ethereum. Out of traditional machine learning models like regression, K-means and Random Forest, we found ARIMA to be the best among them. The intention was to compare a traditional machine learning model with a recurrent neural network model to assess performance improvements. The models were tested on different subsets of the data. The hyperparameters were tuned with the intention of obtaining the least root mean squared error value.

ARIMA models are generally used for time series analysis and forecasting. The model that is used on the temporo sequential data under consideration. The data is transformed into a stationary time series with the features obtained as inputs to predict the log return. At every time t, we train a model to predict a log return at time t.

Recurrent neural networks have been used for time series data prediction problems. The major advantage offered by RNNs is their ability to allow the networks to use data from previous passes in the loops, which acts as memory [21]. The problem that is faced by traditional RNN architectures is the fact that neural networks often do not perform to their optimal level unless the underlying time series data is very long and from a very stable system [4]. The reason for this is that there is not enough training data available and non-stationarity in the data will not be handled adequately [28].

Here, $x_0 ... x_t$ represent the inputs a different time periods. The inputs pass through hidden states present in every A, which represent the information contained by the network at any given time. This is based on the current input x_t and the previous hidden states $h_0 \dots h_t$. In this way, the current information present is dependent on previous information that has been gathered.The process of carrying forward the memory mathematically can be described by the equation $h_t = \phi(W x_t + U h_t - 1)$. The hidden state at time t , h_t is a function of input x_t and a weight matrix W. The weight matrix is added to the previous hidden state $h_t - 1$ multiplied by transition matrix U.The errors generated will be used to adjust and determine the degree of importance of the weight matrices consisting of current and previous

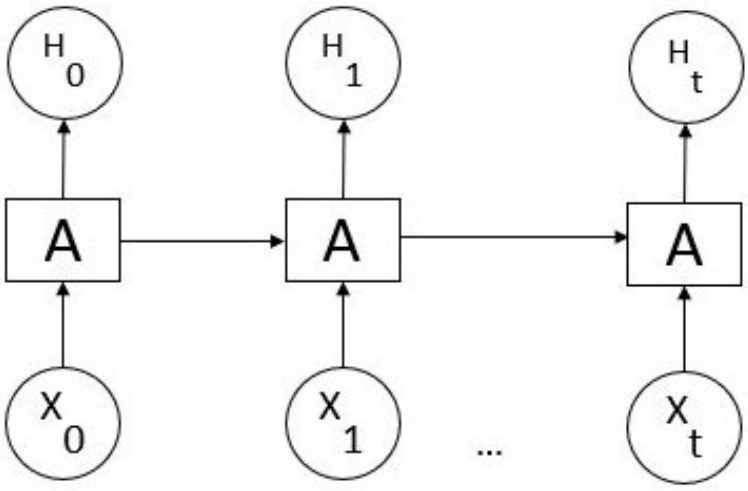

Figure 3: An unrolled RNN loop

hidden states.

We considered a recurrent neural network with long short term memory cells. Neural networks have been known to work well with seasonal and cyclic patterns in time sensitive data, due to their ability to estimate linear and non-linear functions [42]. LSTM networks were introduced by Hochreiter and Schmidhuber [27]. We consider four different types of cells to input into the layers - namely , a basic RNN cell , a basic LSTM cell , an LSTM cell with peephole connections and GRU cells. GRU cells were first introduced in 2014 [12]. The hyperparameters were tuned on an iterative basis until the lowest RMSE value was achieved. RMSE has the advantage of penalizing large errors [8], which is a useful consideration when the log return is being predicted, as cryptocurrency values can fluctuate wildly [40]. Peephole connections [20] and vanilla LSTM cells were considered and were found to performed worse when compared to GRU cells.

4.1 Architecture and hyperparameter description

We use the Tensorflow [3] software library. The dataset is loaded and normalized, after which is represented as an array. Then, different sequences of the data of a fixed length are created and converted back into an array

representation. The sequence length being used here is 11. The dataset is then split into training, validation and test sets in an 80 / 10 / 10 ratio. A 2 layer RNN with 128 neurons in each layer is considered. GRU cells are added to the layers with Leaky Relu as the activation function. The layers are combined into a multi RNN cell and fed into the Tensorflow session in batches of size 64. The intention of the process is to reduce the RMSE, for which an Adam optimizer [31] is used along with a learning rate of 0.01. The setup is run for a total of 175 epochs.

5 Results

For understanding the importance of adding various kinds of network features to the dataset, we consider the performance of the models on three types of datasets – the dataset with just the log return and price values (dataset 1), the dataset with log return , price and summary graph features (dataset 2) and the dataset with log return values, price values , summary graph features and additional graph features mentioned in section 3.3(dataset 3). The RMSE values for the test set for different model and dataset combinations are reported in table 6. We also consider a series of baseline models to compare the performance

We consider the ARIMA model to be our baseline model. We achieved an RMSE of 0.011, which is the threshold for the neural network model to beat.

Naïve forecast	Simple exponential smoothing	ARIMA	Random walk
0.012	0.013	0.012	0.0002

Table 5: Results

5.1 Best Performance

The best performance is seen by the random walk model with an RMSE of 0.0002. It is important to consider the fact that while random walk methods

	Dataset1	Dataset2	Dataset3
Basic RNN cell	0.017	0.131	0.033
Basic LSTM cell	0.021	0.101	0.035
LSTM cell with Peephole connections	0.021	0.085	0.029
GRU cell	0.02	0.091	0.026

Table 6: Results

are generally known to perform extremely well with time series data that is small , the performance of these models for price prediction models is not known to perform extremely well with out of sample observations.

For the baseline models, the RMSE outputs for naive forecast , simple exponential smoothing and ARIMA have no noticeable difference between the RMSE values. We have observed that while the RMSE for these models is low, they do not do a good job at predicting extreme events(i.e., observations with a very high or low log return value).

For the RNN LSTM models , we see that the RMSE does keep decreasing as more variables are added to the LSTM cells. The change in RMSE is highest for dataset 3 , which leads us to believe that further hyperparameter manipulation does present the possibility of a reduction in RMSE. We also observed that the RNN LSTM models are generally better at accounting for extreme events. It is important to understand the fact that recent upticks in the price of Ethereum have resulted in outliers in the data. Additionally, due to the relatively small size of the database, overfitting was a major issue and it was important to maintain a tradeoff between reducing the RMSE for the training and the test set, which can lead to the model being thrown off. With the availability of more data over time, we anticipate that the model will get better at log return prediction.

6 Conclusion

The task of predicting the log return for Ethereum is significant. While both the methods have significant results when compared to other machine learning models, the recurrent neural network model has shown the best results. We expect better architectures and models of neural networks to be available in the future along with a dataset that is bigger, as the amount

of time that Ethereum has existed for increases. We are encouraged by the results thus far and will continue to work on the areas mentioned in section 7.

7 Future Work

There is a clear relationship between the addition of more features and the reduction of the RMSE value. This leads us to believe that adding more features is an area that can be pursued. Specifically, three areas of adding features can be worth considering. Firstly, adding features that would be more commonly seen in stock price prediction problems [7]. Examples of this include running averages and index values. Secondly, adding more network based features that can be extracted from the transaction data. Examples of this include adding data on network motifs and specific node characteristics. Lastly, adding features that are derived from users. This includes adding sentiment scores and forum conversation data. The same study can also be replicated for other cryptocurrencies in the future.

References

[1] Explore decentralized applications (projects built on ethereum), 2018.

[2] Web3 javascript app api for 0.2x.x. `https://github.com/ethereum/wiki/wiki/JavaScript-API`, 2018.

[3] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[4] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. Forecasting across time series databases using long short-term memory networks on groups of similar series. *arXiv preprint arXiv:1710.03222*, 2017.

[5] Annika Baumann, Benjamin Fabian, and Matthias Lischke. Exploring the bitcoin network. In *WEBIST (1)*, pages 369–374, 2014.

[6] Zehra Çataltepe, Savaş Özer, and Vahide Unutmaz Barın. Feature selection for price change prediction.

[7] Zehra Çataltepe, Savaş Özer, and Vahide Unutmaz Barın. Feature selection for price change prediction.

[8] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

[9] Stephen Chan, Jeffrey Chu, Saralees Nadarajah, and Joerg Osterrieder. A statistical analysis of cryptocurrencies. *Journal of Risk and Financial Management*, 10(2):12, 2017.

[10] Matthew Chen, Neha Narwal, and Mila Schultz. Predicting price changes in ethereum.

[11] Jonathan Chiu and Thorsten V Koeppl. The economics of cryptocurrencies–bitcoin and beyond. 2017.

[12] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[13] Coindesk. Ethereum price.

[14] Coindesk. *State of Blockchain 2018*. 2017.

[15] Michael Crosby, Pradan Pattanayak, Sanjeev Verma, and Vignesh Kalyanaraman. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2:6–10, 2016.

[16] Chris Dannen. *Introducing Ethereum and Solidity*. Springer, 2017.

[17] Christos Faloutsos, Kevin S Mccurley, and Andrew Tomkins. Connection subgraphs in social networks. In *SIAM International Conference on Data Mining, Workshop on Link Analysis, Counterterrorism and Security*, 2004.

[18] Yaya J Fanusie and Tom Robinson. *Bitcoin Laundering: An Analysis of Illicit Flows into Digital Currency Services*. Jan 2018.

[19] Sean Foley, Jonathan Karlsen, and Tālis J Putniņš. Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? 2018.

[20] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[22] Alex Greaves and Benjamin Au. Using the bitcoin transaction graph to predict the price of bitcoin. *No Data*, 2015.

[23] Tian Guo and Nino Antulov-Fantulin. Predicting short-term bitcoin price fluctuations from buy and sell orders. *arXiv preprint arXiv:1802.04065*, 2018.

[24] Aric Hagberg, Dan Schult, Pieter Swart, D Conway, L Séguin-Charbonneau, C Ellison, B Edwards, and J Torrents. Networkx. high productivity software for complex networks. *Webová strá nka https://networkx. lanl. gov/wiki*, 2013.

[25] KAREEM HEGAZY and SAMUEL MUMFORD. Comparitive automated bitcoin trading strategies.

[26] Ronald Heijmans, Richard Heuver, Clement Levallois, and Iman van Lelyveld. Dynamic visualization of large transaction networks: the daily dutch overnight money market. 2014.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[28] Rob J Hyndman, Earo Wang, and Nikolay Laptev. Large-scale unusual time series detection. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pages 1616–1619. IEEE, 2015.

[29] Aggelos Kiayias, Andrew Miller, and Dionysis Zindros. Non-interactive proofs of proof-of-work, 2017.

[30] Young Bin Kim, Jun Gi Kim, Wook Kim, Jae Ho Im, Tae Hyeong Kim, Shin Jin Kang, and Chang Hun Kim. Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PloS one*, 11(8):e0161197, 2016.

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] Dániel Kondor, István Csabai, János Szüle, Márton Pósfai, and Gábor Vattay. Inferring the interplay between network structure and market effects in bitcoin. *New Journal of Physics*, 16(12):125003, 2014.

[33] Dániel Kondor, Márton Pósfai, István Csabai, and Gábor Vattay. Do the rich get richer? an empirical analysis of the bitcoin transaction network. *PloS one*, 9(2):e86197, 2014.

[34] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 2017.

[35] Isaac Madan, Shaurya Saluja, and Aojia Zhao. Automated bitcoin trading via machine learning algorithms. *URL: http://cs229. stanford. edu/proj2014/Isaac% 20Madan*, 20, 2015.

[36] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[37] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[38] Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association*, 96(455):1077–1087, 2001.

[39] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer, 2013.

[40] Anthony Serapiglia and PA Latrobe. Cryptocurrencies: Technology empowered integration of information systems enabling currency without borders. In *Proceedings of the Information Systems Educators Conference ISSN*, volume 2167, page 1435. Citeseer, 2014.

[41] Evita Stenqvist and Jacob Lönnö. Predicting bitcoin price fluctuation with twitter sentiment analysis, 2017.

[42] Zaiyong Tang, Chrys de Almeida, and Paul A Fishwick. Time series forecasting using neural networks vs. box-jenkins methodology. *Simulation*, 57(5):303–310, 1991.

[43] Northwestern University. *The structure of transportation networks*. 1962.