Analysis and Prediction of Cryptocurrencies and Ethereum Smart Contracts

Coursework for Data Analytics: ECS784P

Queen Mary University of London

April 2018

Mohamed Baddar
m.baddar@se17.qmu.ac.uk

Muhammad Omar Waqar
m.waqar@se17.qmul.ac.uk

Daniyal Asad Chughtai
d.chughtai@se17.qmul.ac.uk

Ahmed Waseef
a.waseef@se17.qmul.ac.uk

Contents

Abstract

Last year witnessed the rise and decline of cryptocurrencies led by Bitcoin and Ethereum (Wikipedia). Many believe the blockchain (Wikipedia), and cryptocurrencies (Wikipedia) as its primary use case, will change the world. Whereas many others believe it is the new dot com bubble. In this coursework we have collected a huge dataset of the Ethereum blockchain, analyzed smart contracts on a subset of our collected data. And finally done more analysis on the correlation between Bitcoin, the first and most valued cryptocurrency, Ether and other currencies. We extended the concept of (Sifr Data, 2018) which publish the correlation on the last number of days to the weekly and monthly correlation. Which, we believe, makes more sense due to the huge volatility of this market.

Problem statement

We wanted to build a proof of concept to answer the following questions:

1. Are cryptocurrencies consistently correlated to Bitcoin? What is the point of holding a portfolio of different currencies if that is the case?
2. Can we predict the price of Bitcoin?
3. What is the market share of smart contracts hosted in the Ethereum blockchain?
4. How is the trend of smart contracts changing with time compared to normal transactions in Ethereum blockchain?
5. What are some of the top smart contracts on Ethereum blockchain in terms of value?

Background

Many attempts have been made to create different forms of electronic currencies or generally means of exchange that do not necessarily involve banks or a trusted intermediary. David Chaum (Wikipedia), an American cryptographer have founded DigiCash (Wikipedia) in the Netherlands, back in 1989. An innovation way ahead of its time (Amazon was founded in 1994. Ecommerce and online payment was on its infancy). The Company later filed for bankruptcy and was sold for assets. Even Paypal (Throwback Thursday: PayPal's Biggest Days In History, 2015) first adopted peer-to-peer money before they discarded the idea later.

Many centralized peer-to-peer money innovations were founded and faded for either lack of adoption or regulatory busts followed by government closures.

Then comes Bitcoin.

Inspired by Nick Szabo’s Bit Gold. An idea that he never implemented. A person or a group go by the name Satoshi Nakamoto invented Bitcoin in 2008. The first peer-to-peer currency that does not have a corporation for governments to target or a data center to seize. That was the digital anarchist dream come true.

Bitcoin was the first adopted implementation of a currency that:

1. Is Peer-to-Peer
2. Does not need a trusted intermediary to settle transactions
3. Decentralized
4. AND: Cannot be double-spent (Wikipedia)

No one have achieved those properties combined before. And that is why Bitcoin is here to stay.

Fast forward to 2015. Ethereum was born.

Bitcoin is a cryptocurrency built on blockchain. Ethereum is a distributed computing open platform and a public blockchain. Where many other companies host their own cryptocurrencies (known as tokens) and smart contracts. It also has a native cryptocurrency known as Ether. This distinction between the currency and the network is sometimes overlooked.

Ethereum created a new era for the crypto world by enabling anyone to create their own currency. And more remarkably by inventing what is known as DAO: Decentralized Autonomous Organizations.

This did not go without pain: The first DAO managed to gather 12.7m Ether (worth around $150M at the time), making it the biggest crowdfund ever (Madeira, 2018). It had been later hacked by an unknown attacker who managed to drain 3.6m Ether (worth around $70M at the time).

The ICO craze

Initial Coin Offering (ICO) is an unregulated means by which funds are raised for a new cryptocurrency venture. An Initial Coin Offering (ICO) is used by startups to bypass the rigorous and regulated capital-raising process required by venture capitalists or banks. In an ICO campaign, a percentage of the cryptocurrency is sold to early backers of the project in exchange for legal tender or other cryptocurrencies, but usually for Bitcoin (Investopedia).

In 2017. Startups and projects raised $5.6 billion through initial coin offerings (ICOs), according to a new report (Williams-Grut, 2018). That was the biggest catalyst to the hyperinflation of bitcoin and other cryptocurrencies.

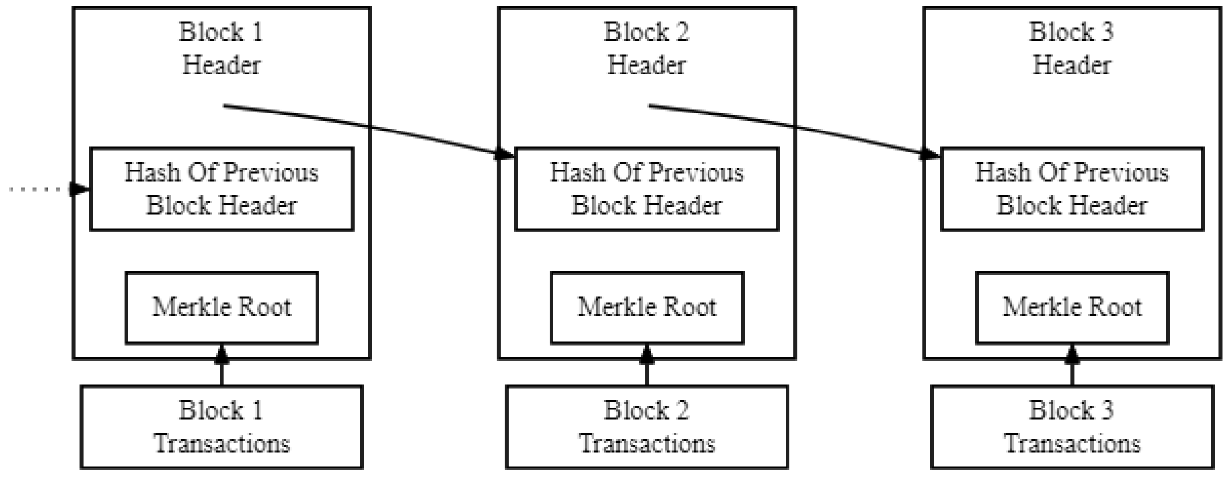
Hence, we chose this topic for our coursework!

The blockchain

This section is a brief technical overview of the Ethereum blockchain. We will only explain some definitions and concepts necessary for the reader to follow our analysis. For more information please refer to the Ethereum wiki[1] and the Ethereum white paper[2]

What is a blockchain

A blockchain is an ordered and timestamped record of transactions. This system is used to protect against double spending and modification of previous transaction records. (bitcoin.org, 2018)



Simplified Bitcoin Block Chain

Ethereum Turing Completeness (Castillo, 2018) have made it necessary to extend this concept (Buterin, 2014). More information on (Stack Exchange, 2017)

Ethereum Accounts

In Ethereum, the state is made up of objects called "accounts", with each account having a 20-byte address and state transitions being direct transfers of value and information between accounts. An Ethereum account contains four fields:

- The nonce, a counter used to make sure each transaction can only be processed once
- The account's current ether balance

[1] https://github.com/ethereum/wiki/wiki
[2] https://github.com/ethereum/wiki/wiki/White-Paper

- The account's contract code, if present
- The account's storage (empty by default) (Ray, 2018)

Smart Contracts

Smart contracts are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein exist across a distributed, decentralized blockchain network. Smart contracts permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a central authority, legal system, or external enforcement mechanism. They render transactions traceable, transparent, and irreversible. (Investopedia)

Decentralized Autonomous Organizations

An application of the Ethereum decentralized application platform. The general concept of a "decentralized autonomous organization" is that of a virtual entity that has a certain set of members or shareholders which, perhaps with a 67% majority, have the right to spend the entity's funds and modify its code. The members would collectively decide on how the organization should allocate its funds (Ray, 2018)

Leveldb

The database used to store the Ethereum blockchain. (Ghemawat & Dean)

Geth

The official go implementation of the Ethereum protocol. One of many Ethereum client implementations. (Ethereum.org, 2018)

Transaction

The term "transaction" is used in Ethereum to refer to the signed data package that stores a message to be sent from an externally owned account. Transactions contain:

- The recipient of the message
- A signature identifying the sender
- The amount of ether to transfer from the sender to the recipient
- An optional data field
- A STARTGAS value, representing the maximum number of computational steps the transaction execution is allowed to take
- A GASPRICE value, representing the fee the sender pays per computational step (Ray, 2018)

Refer to Appendix D: Output overview for a sample JSON transaction

EVM

Ethereum Virtual Machine (EVM) can be thought of as a large decentralized computer containing millions of objects, called "accounts", which has the ability to maintain an internal database, execute code and talk to each other. (Liu, 2018)

Solidity

Solidity is a contract-oriented, high-level language for implementing smart contracts. It was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM). (The Ethereum Foundation, 2018). Appendix D: Output overview contains a sample solidity code

Mining

The word mining originates in the context of the gold analogy for crypto currencies. Gold or precious metals are scarce, so are digital tokens, and the only way to increase the total volume is through mining it. This is appropriate to the extent that in Ethereum too, the only mode of issuance post launch is via mining. Unlike these examples however, mining is also the way to secure the network by creating, verifying, publishing and propagating blocks in the blockchain.

Mining Ether = Securing the network = verify computation

Gas Limit/Gas price

Ethereum is the network, also known as the blockchain. Ether (ETH) is the fuel for that network. When you send tokens, interact with a contract, send ETH, or do anything else on the blockchain, you must pay for that computation. That payment is calculated in `Gas` and gas is paid in `ETH`.

The total cost of a transaction (the "TX fee") is the `Gas Limit` * `Gas Price`. (MyEtherWallet, 2018)

ERC20 Token

Similar to how the HTTP protocol defined the internet, ERC20 is a protocol that defines a set of commands that a token should implement. ERC20 is not a technology, software, or piece of code. It is a technical specification. If a token implements the spec, it is an ERC20 token (Siebel, 2017)

ERC20 tokens are most widely used for companies to issue their own cryptocurrencies and smart contract assets.

JSON-RPC

JSON-RPC is a stateless, light-weight remote procedure call (RPC) protocol. Geth also operates as a JSON-RPC server. (Ethereum.org, 2018)

The Data

We wanted to analyze Ethereum as it hosts 91% of token sales at the beginning of 2018 (AminCad, 2018). The next logical step is to find a dataset that contains proportionally sufficient transaction volume to do our analysis on.

Unfortunately, we could not find such a dataset from any online source. The available datasets only track the daily prices against USD.

That was a shocking finding!

We then reasoned that well, someone must have thought about it, so we should easily be able to find some code on github. In that case, all we need to do is just run it and get our dataset.

We found a github repository (Miller, 2017) that extracts the block information and transactions into a MongoDB. We wanted to get a year's worth of data. We first needed a full node (The entire blockchain. More than 5 million blocks, more on that later). That step took a few days to complete. Downloading and verifying about 60 GB worth of data on a Windows 7 machine.

The first attempt to run the code failed!

The code was based on python 2.7. It also uses outdated python and Ethereum modules. After doing some research we managed to replace the outdated modules. One of the core dependencies was pyethereum[3] and, well, it does not run on Windows! It has to be built from source and it has Linux dependencies. So, we ended up with a blockchain on Windows and a code that must run on Linux.

To export the blockchain to Linux we need a machine with at least 60GB of disk space and a couple of days to re-verify transactions again. That was a time we did not have!

The only solution was to keep the blockchain on Windows and run the code on a Linux VM that connects to our Windows box.

Due to time constraint we ruled out using AWS or similar cloud VM. We used a local Virtualbox (About VirtualBox, 2018) Ubuntu 16.04 machine.

We wanted to get the data of around 1 million blocks. The code is designed to crawl from the very beginning. We customized it, and we finally managed to get the code running.

It was communicating with the blockchain via http requests. 1 at a time. The amount of data we needed (around 2.38 m blocks) would have taken more than 660 hours at 1 block/sec. speed! That was the current code capacity.

Again, a hurdle we needed to overcome.

We found a pull request[4] on that repository (the repo owner does not maintain it any more) that claimed a higher throughput. It had a slightly different approach to crawl the chain however it had many defects we needed to fix (that pull request was never merged to the repo master branch. So, no one tested it apart from its creator).

[3] Github. Pyethereum. https://github.com/ethereum/pyethereum
[4] https://github.com/alex-miller-0/Ethereum_Blockchain_Parser/pull/3

We finally got our code running and progressing in a reasonable time. About 90% reduction from the original time estimation!

Data Wrangling accomplished!

Data Exploration: Smart Contracts

The data wrangled in the previous part was stored as a MongoDB document database. To explore it, we had to use a MongoDB server, pymongo (python) package and Studio 3T for MongoDB. MongoDB server was installed and initialized on local machine for ease of use. Initially Studio 3T was used to get a visual representation of the documents in the database, their format and size. Due to the data being too large, a subset of the data was used to formulate the queries that would extract data.

Our first approach was to use pymongo and write code in python to query the required data from the database. This approach worked well on the subset of the data and we were able to extract and save query results in .json files. These files were then used in python for plotting and visualizing the different metrics. However, on the complete dataset, this approach failed because the queries would take too long and either the operation would time out, or the system would run out of memory. Just as an example, a query to extract contract creation transactions from all the blocks in the dataset took about 9 hours to complete. Other queries either took longer or failed after 7-8 hours of operation.

After failed attempts to optimize the queries in various ways, we resorted to using Studio 3T for data extraction. It was able to optimize the queries internally and gave out results in the form of MongoDB documents much quicker. These documents were used for plotting and visualization.

Following is some information about the dataset:

Time span of the data set:	December 2016 – July 2017
Total number of blocks:	1,169,618
Total number of transactions:	26,798,548

The scope of this report is restricted to analyzing smart contracts on the blockchain so our data extraction mainly revolves around them.

A brief explanation of the database queries and the plotting code refer to Appendix B: Data extraction and visualization

Smart contract market share

The market share is determined by the amount of Ether (ETH) traded in transactions involving contract addresses and those with normal addresses.

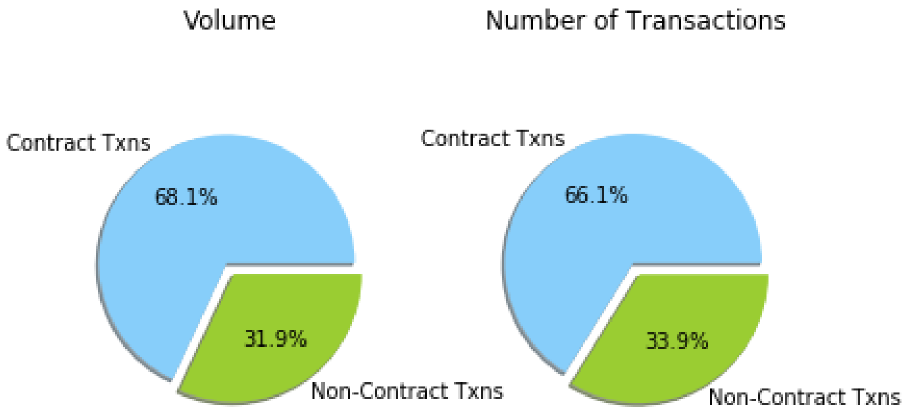


Figure 1: Market Capitalization as of July-2017

The figure above shows that smart contracts hold a 68.1% in terms of ETH value and 66.1% in terms of number of transactions done through smart contracts. Considering Ethereum is the main platform for smart contracts, it is expected to hold a higher share.

Smart Contract Creation Trend

Number of contracts created each month would tell us about how the idea of smart contracts is gaining/losing traction.

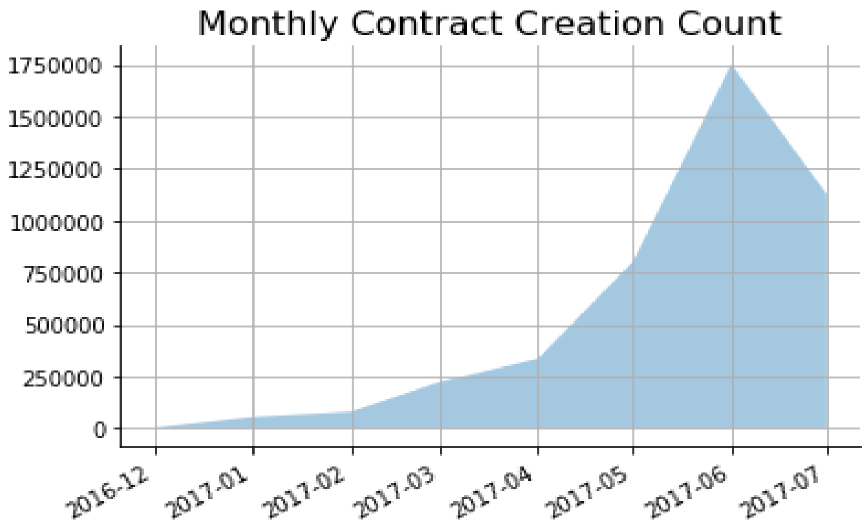


Figure 2: Contracts created each month

Figure above shows a growing trend in the contracts created each month. This growth shows an exponential trend, meaning the smart contracts are growing popular with time. The magnitude of this growth can be explained by the growing popularity of cryptocurrency in general as well as the increasing adoption of smart contracts. As our data starts from December 2016 and ends in July 2017, the values for these months are not accurate due to incomplete data for these months. We did not exclude these months from our plot for the sake of completeness.

Smart contract transaction volume compared to normal transactions

By comparing transaction volumes in monthly time interval, we can get some insights on how the smart contract fare against normal transactions and any changes in the trend if any.

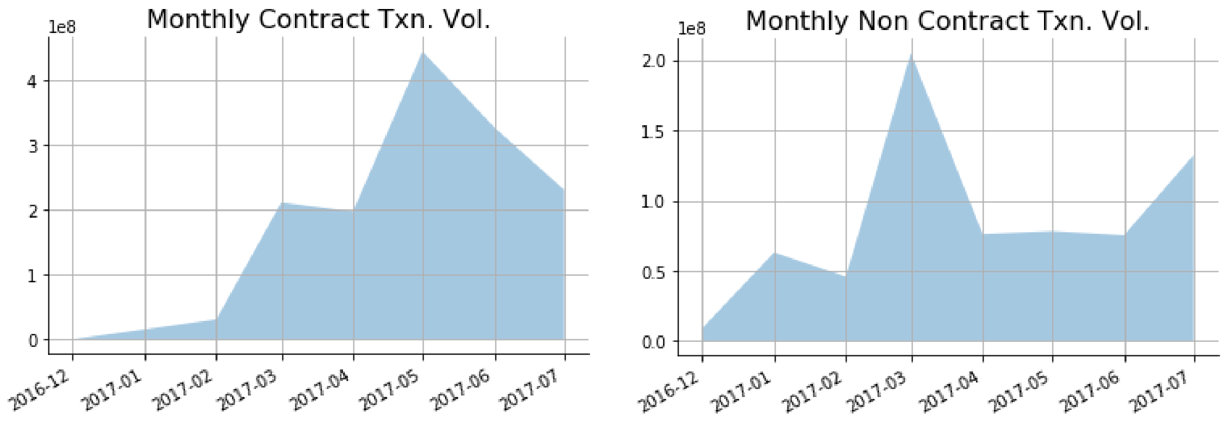


Figure 3: Comparison between smart contract and normal transaction volumes with time

From the y-axis scale it is clear that smart contracts have gone up considerably in comparison to normal transactions. Up until February 2017 normal transactions were higher in volume then smart contracts. In March 2017 smart contracts superseded normal transactions in terms of volume and since they have been on the rise. Another behavior shown in the graphs is that during the period March-July 2017, the smart contracts show a peak in the May 2017 while normal transactions dipped and remain low throughout this time. This shows that there was more activity in smart contracts then normal transactions although this was a period where cryptocurrency was very popular.

Notable Smart Contracts

The following table shows the top smart contracts with high volume of transactions:

Rank	Address	Name/Tag	Volume	No. of Transactions
1	0x209c4784ab1e8183cf58ca33cb740efbf3fc18ef	Poloniex_2	12362766.225237455	587990
2	0x7727e5113d1d161373623e5f49fd568b4f543a9e	Bitfinex_Wallet2	12072289.758089611	64061
3	0xfa52274dd61e1643d2205169732f29114bc240b3	Kraken split contract	9981656.455467101	211251
4	0x6fc82a5fe25a5cdb58bc74600a40a69c065263f8	(Possible Spam)	4213587.620218947	29604
5	0xe94b04a0fed112f3664e45adb2b8915693dd5ff3	Bittrex_2	3680807.621031744	276257

The above table show some of the top contract addresses as on July 2017. The names or tags have been taken from an online Ethereum explorer https://etherscan.io/.

As can be seen from the table, all the top 5 smart contracts, excluding one, belong to cryptocurrency exchanges. This could be because of high amount of trading in cryptocurrencies during this period, a lot of transactions have been happening through these cryptocurrencies.

Challenges with Data Extraction

Extracting data and exploring it has its challenges. In our case after weeks of effort to extract meaningful fields from the dataset, we discovered that some of the dataset entries were not accurate. For example, some of the transactions were wrongly classified as smart contract transactions. Ideally, we were supposed to fix our crawler code and re-crawl the data to get the data right but due to time constraints and taxing process of data crawling we had to settle for filtering out the anomalies at the data extraction stage. Overall these anomalies have not much of an effect on our results as they result in the same conclusion.

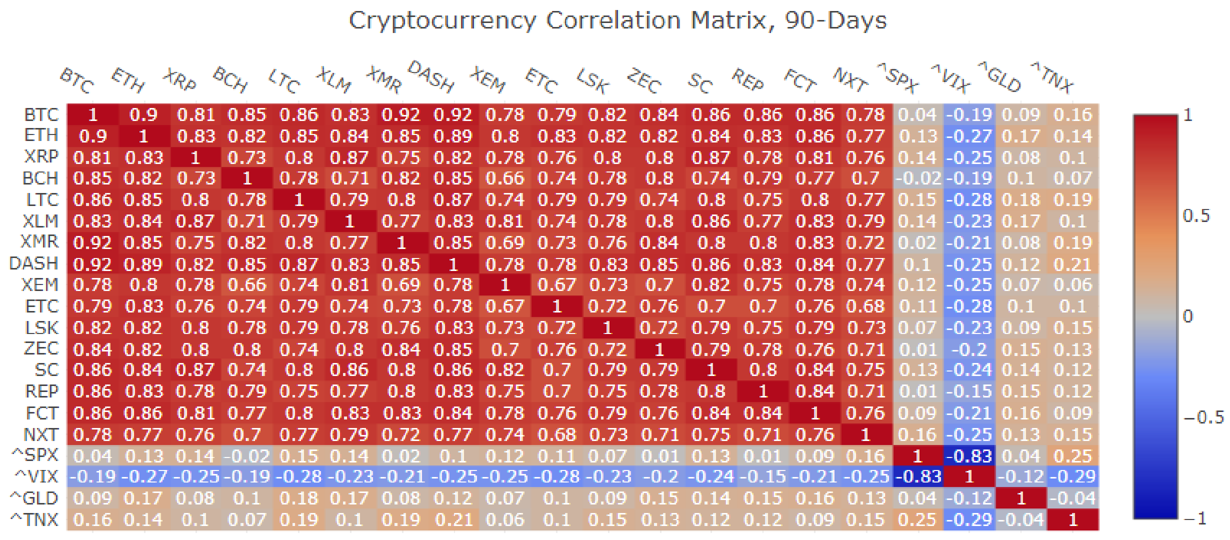
Further Analysis: Correlation and Prediction

In our quest to understand the hugely volatile price movements of cryptocurrencies. We thought that understanding the correlation between Bitcoin and Ethereum and applying some time series analysis is the way to go. We had the impression that every cryptocurrency (There are more than a thousand of them!) is correlated with Bitcoin. And when Bitcoin sinks, every other currency follows.

We perceived Ethereum and Bitcoin should have different fundamental drivers and hence we did not buy into the above common presumption. Bitcoin is the most traded currency and the one with the most trading pairs on every crypto Exchange out there. Whereas Ethereum is an open platform and should be driven by its tokens. Whenever one interesting project succeeds. That adds to the entire network value and should reflect on price.

We also know that part of the correlation between Ether and Bitcoin (BTC) is that many tokens cannot be directly liquidated (sold for a fiat currency such as GBP). So if a token holder wants to cash out. They first need to sell it for Ether, then from Ether to either fiat or BTC depending on their location. In that context BTC is the most liquid crypto.

We looked at the cryptocurrency correlation matrix of (Sifr Data, 2018) and thought that 90 days is too much time in the crypto world. We should do a finer analysis in order to get any insight out of correlation.



	BTC	ETH	XRP	BCH	LTC	XLM	XMR	DASH	XEM	ETC	LSK	ZEC	SC	REP	FCT	NXT	^SPX	^VIX	^GLD	^TNX
BTC	1	0.9	0.81	0.85	0.86	0.83	0.92	0.92	0.78	0.79	0.82	0.84	0.86	0.86	0.86	0.78	0.04	-0.19	0.09	0.16
ETH	0.9	1	0.83	0.82	0.85	0.84	0.85	0.89	0.8	0.83	0.82	0.82	0.84	0.83	0.86	0.77	0.13	-0.27	0.17	0.14
XRP	0.81	0.83	1	0.73	0.8	0.87	0.75	0.82	0.78	0.76	0.8	0.8	0.87	0.78	0.81	0.76	0.14	-0.25	0.08	0.1
BCH	0.85	0.82	0.73	1	0.78	0.71	0.82	0.85	0.66	0.74	0.78	0.8	0.74	0.79	0.77	0.7	-0.02	-0.19	0.1	0.07
LTC	0.86	0.85	0.8	0.78	1	0.79	0.8	0.87	0.74	0.79	0.79	0.74	0.8	0.75	0.8	0.77	0.15	-0.28	0.18	0.19
XLM	0.83	0.84	0.87	0.71	0.79	1	0.77	0.83	0.81	0.74	0.78	0.8	0.86	0.77	0.83	0.79	0.14	-0.23	0.17	0.1
XMR	0.92	0.85	0.75	0.82	0.8	0.77	1	0.85	0.69	0.73	0.76	0.84	0.8	0.8	0.83	0.72	0.02	-0.21	0.08	0.19
DASH	0.92	0.89	0.82	0.85	0.87	0.83	0.85	1	0.78	0.78	0.83	0.85	0.86	0.83	0.84	0.77	0.1	-0.25	0.12	0.21
XEM	0.78	0.8	0.78	0.66	0.74	0.81	0.69	0.78	1	0.67	0.73	0.7	0.82	0.75	0.78	0.74	0.12	-0.25	0.07	0.06
ETC	0.79	0.83	0.76	0.74	0.79	0.74	0.73	0.78	0.67	1	0.72	0.76	0.7	0.7	0.76	0.68	0.11	-0.28	0.1	0.1
LSK	0.82	0.82	0.8	0.78	0.79	0.78	0.76	0.83	0.73	0.72	1	0.72	0.79	0.75	0.79	0.73	0.07	-0.23	0.09	0.15
ZEC	0.84	0.82	0.8	0.8	0.74	0.8	0.84	0.85	0.7	0.76	0.72	1	0.79	0.78	0.76	0.71	0.01	-0.2	0.15	0.13
SC	0.86	0.84	0.87	0.74	0.8	0.86	0.8	0.86	0.82	0.7	0.79	0.79	1	0.8	0.84	0.75	0.13	-0.24	0.14	0.12
REP	0.86	0.83	0.78	0.79	0.75	0.77	0.8	0.83	0.75	0.7	0.75	0.78	0.8	1	0.84	0.71	0.01	-0.15	0.15	0.12
FCT	0.86	0.86	0.81	0.77	0.8	0.83	0.83	0.84	0.78	0.76	0.79	0.76	0.84	0.84	1	0.76	0.09	-0.21	0.16	0.09
NXT	0.78	0.77	0.76	0.7	0.77	0.79	0.72	0.77	0.74	0.68	0.73	0.71	0.75	0.71	0.76	1	0.16	-0.25	0.13	0.15
^SPX	0.04	0.13	0.14	-0.02	0.15	0.14	0.02	0.1	0.12	0.11	0.07	0.01	0.13	0.01	0.09	0.16	1	-0.83	0.04	0.25
^VIX	-0.19	-0.27	-0.25	-0.19	-0.28	-0.23	-0.21	-0.25	-0.25	-0.28	-0.23	-0.2	-0.24	-0.15	-0.21	-0.25	-0.83	1	-0.12	-0.29
^GLD	0.09	0.17	0.08	0.1	0.18	0.17	0.08	0.12	0.07	0.1	0.09	0.15	0.14	0.15	0.16	0.13	0.04	-0.12	1	-0.04
^TNX	0.16	0.14	0.1	0.07	0.19	0.1	0.19	0.21	0.06	0.1	0.15	0.13	0.12	0.12	0.09	0.15	0.25	-0.29	-0.04	1

Figure 4 Sifr Data 90 Days Crypto Correlation Matrix. As of 12/4/2018

We found a great Jupyter notebook by (Ma, 2018) that replicate the correlation matrix above on even a longer time interval. Along with other very useful tools that we based this part of our analysis on.

The code uses a Kaggle dataset (Vent, 2018). It tracks the daily price of 13 cryptocurrencies since 22/2/2018. It contains historic open, high, low, close, trading volume and market cap. We adopted the 7 currencies she picked, possibly by highest market cap at the time of coding. And we eventually focused on BTC and Ether.

Note: The analysis would have been more insightful if we had hourly prices.

Since this is a preprocessed data source. We thankfully did not have any issue loading the data. Refer to Appendix C: Analytics Jupyter Notebook Code Overview for a sample of the dataframe.

Market Cap

The market cap for Ethereum came second only to Bitcoin, going to third number for a short period of time in January 2018, then rapidly bouncing back. We can see the huge boom and bust of BTC followed by all others at the end of 2017.

It is worth noting that the majority of highly traded Ether token had a delayed trend of around one month since the downtrend of BTC started. Most have peaked in January and started falling.

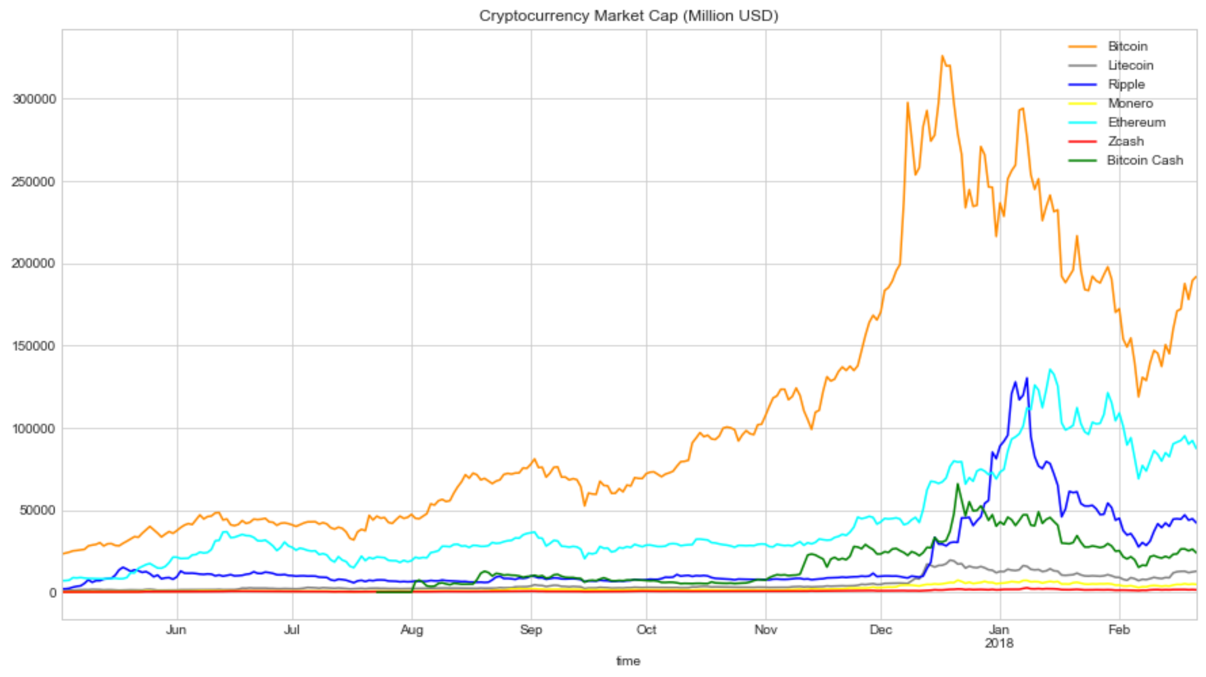


Figure 5 Crypto Market Cap since May 2017

Transaction Volume

The transaction volume graph is denoted in million. The graph shows that Bitcoin is by far the leading cryptocurrency in terms of volume traded. Interestingly, there are some points, where Ethereum overthrew Bitcoin to be the leading cryptocurrency in terms of volume during the months of June till August 2017. We can see the spikes in the graph (denoting a rise in volume) for Bitcoin are way ahead of every other cryptocurrency. After December 2017, the gap between Bitcoin and other cryptocurrencies widened and no one could catch up in terms of volume.

Then came the bust. BTC is trading at less than $8000 at the time of writing.

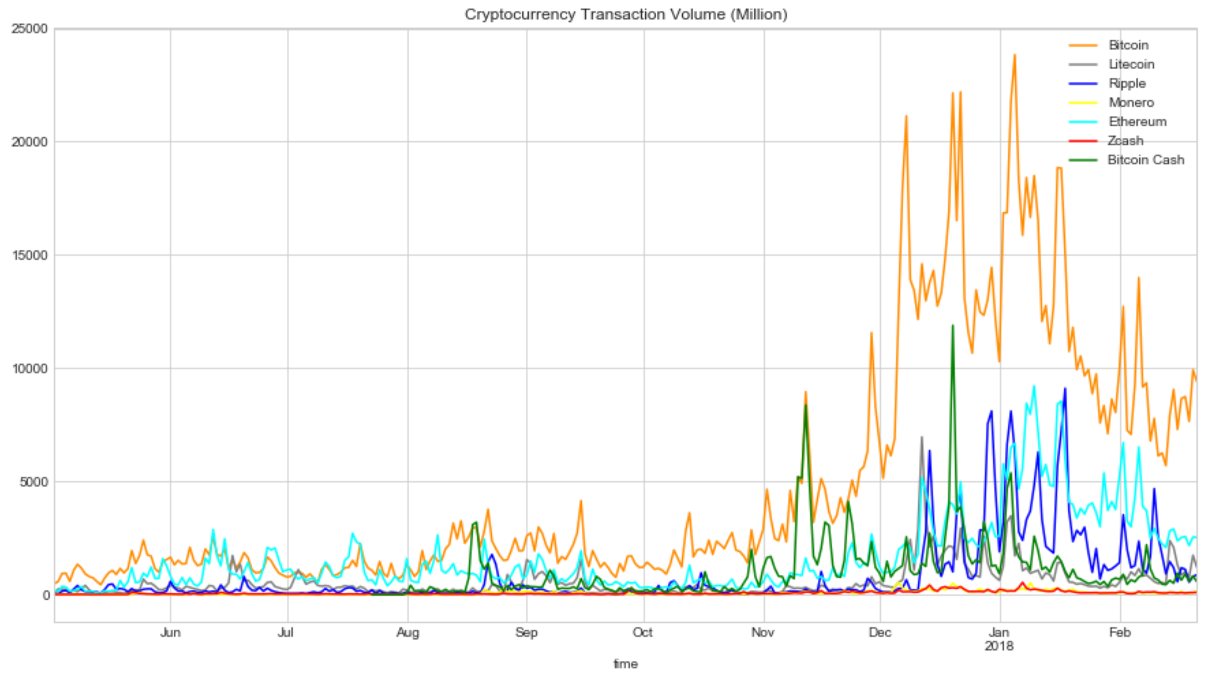


Figure 6 Crypto Transaction Volume

Price Spread

This chart tracks the differences of daily highest and lowest price of the top 7 currencies since 2017. Remarkably, Bitcoin had a price spread of 4000 USD, which could have given investors a chance to earn this amount in a single day for each BTC they trade. Just two out of the seven cryptocurrencies managed to touch 1000 USD price spread. Ethereum saw an increase in price spread after December 2017

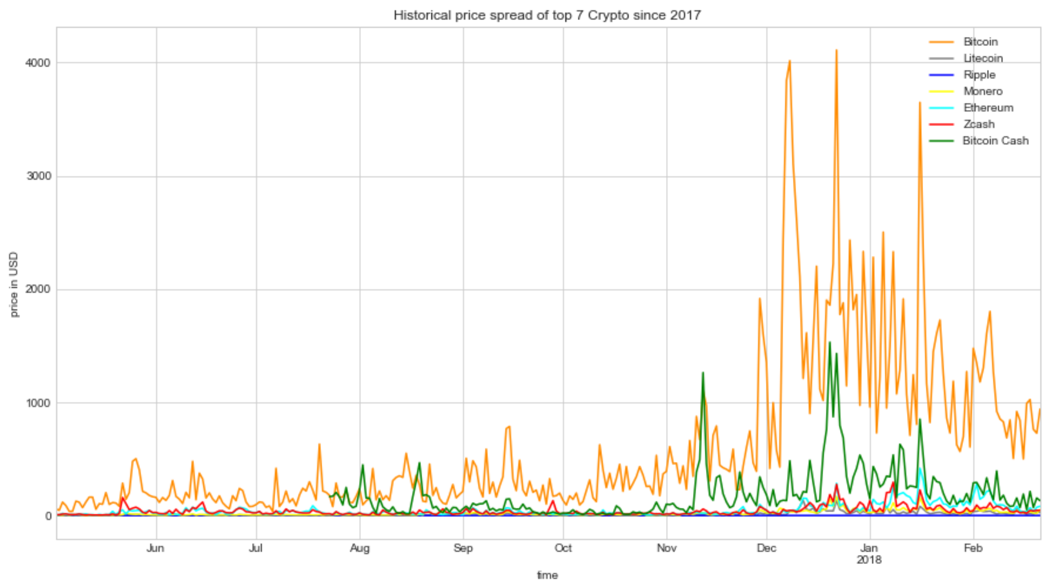


Figure 7 Crypto Price Spread: Since January 2017

Historical daily average prices

The daily average price was calculated by adding up the highest price, lowest price, opening price and the closing price, then dividing the sum by 4. Fluctuation could be a measure of volatility as this averaging should work as a smoothing operator. The plot shows how Bitcoin is the highest in absolute value and volatility. Ether's average peaked in January 2018. That was in line with the token peak mentioned earlier.

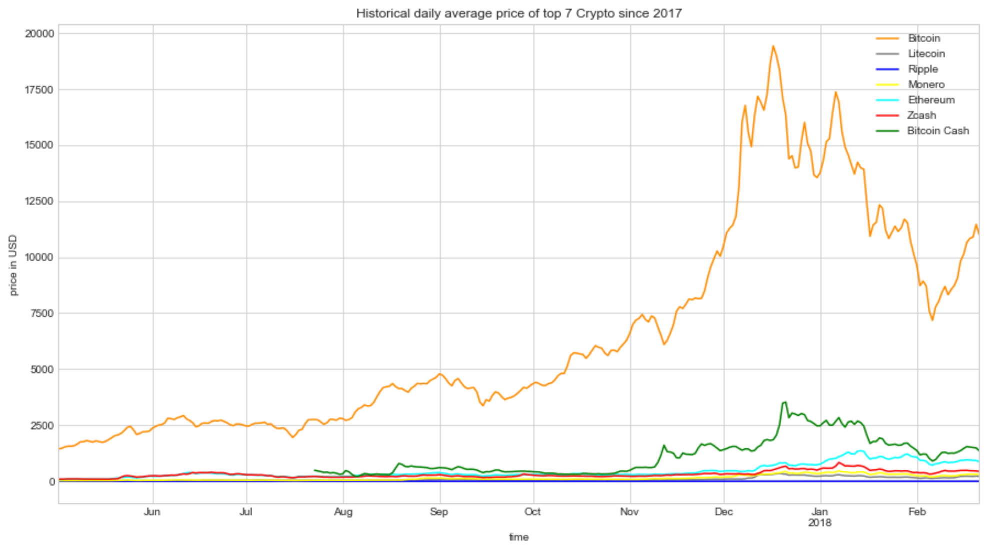


Figure 8 Historical daily average prices since 2017

More smoothing applied by plotting the 5 days moving average of BTC and ETH. We can see how Ether looks flat compared to BTC except in January.

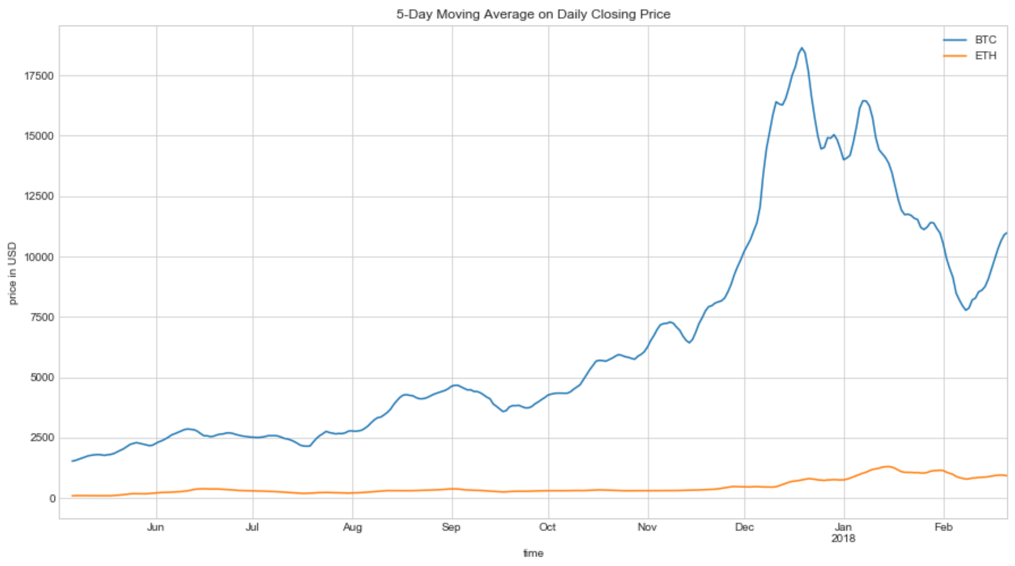


Figure 9 BTC and Ether 5 days moving average

Ether alone follows a very similar trend.

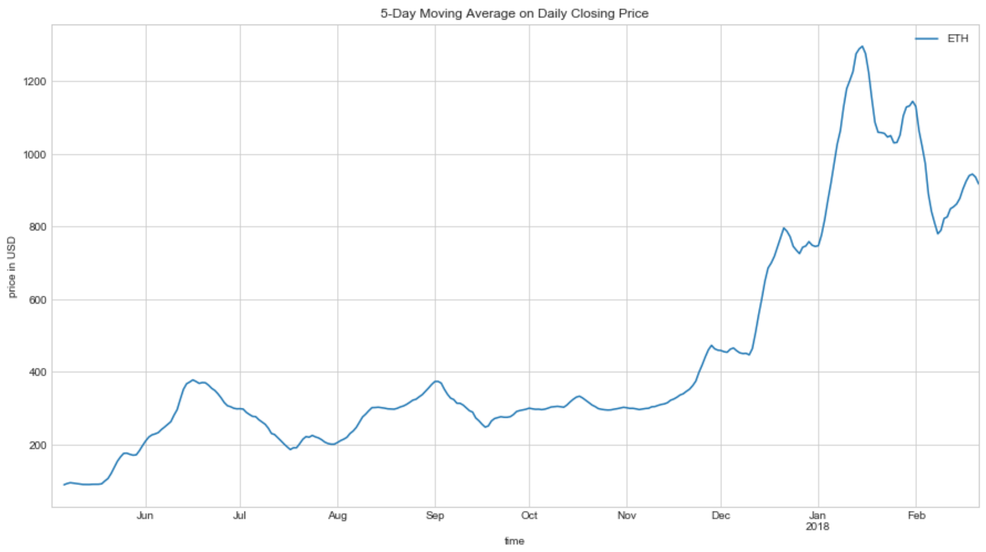
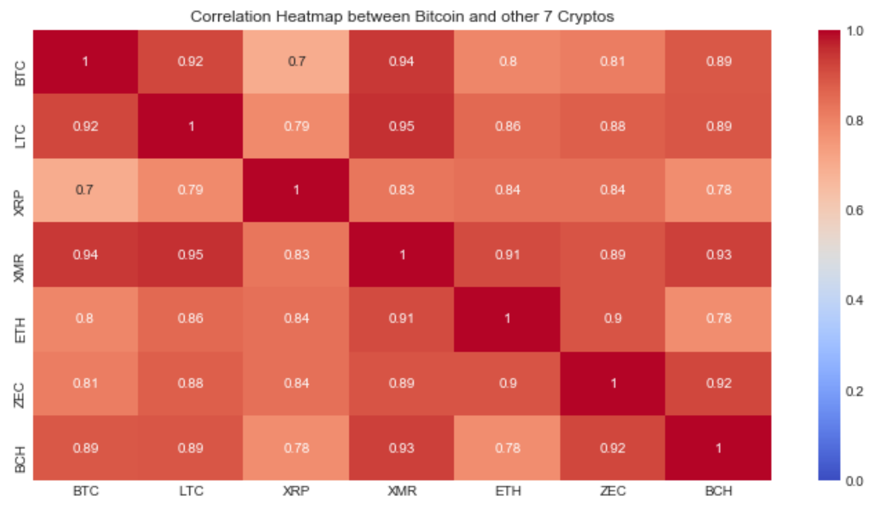


Figure 10 Ether 5 days moving average

Correlation

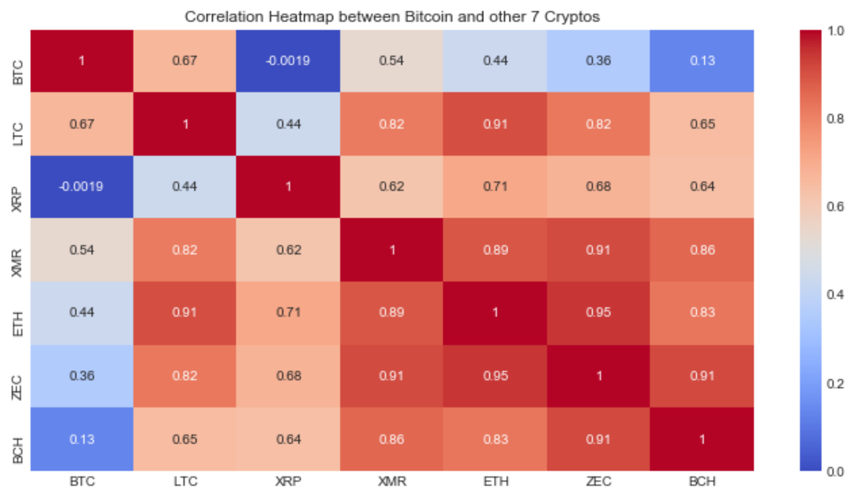
We started by looking at the Pearson correlation heatmap of the 7 currencies for the entire analysis period. We trimmed the data to start from May 2017. The long timespan tells us that the correlation between Ethereum and Bitcoin is quite high (0.8).



	BTC	LTC	XRP	XMR	ETH	ZEC	BCH
BTC	1	0.92	0.7	0.94	0.8	0.81	0.89
LTC	0.92	1	0.79	0.95	0.86	0.88	0.89
XRP	0.7	0.79	1	0.83	0.84	0.84	0.78
XMR	0.94	0.95	0.83	1	0.91	0.89	0.93
ETH	0.8	0.86	0.84	0.91	1	0.9	0.78
ZEC	0.81	0.88	0.84	0.89	0.9	1	0.92
BCH	0.89	0.89	0.78	0.93	0.78	0.92	1

Figure 11 Pearson correlation heatmap of the analyzed 7 cryptocurrencies. May 2017 to February 2018

We did not believe the above was a true measure. That period was too long to represent market dynamics. We then computed the heatmap on a shorter timespan.



	BTC	LTC	XRP	XMR	ETH	ZEC	BCH
BTC	1	0.67	-0.0019	0.54	0.44	0.36	0.13
LTC	0.67	1	0.44	0.82	0.91	0.82	0.65
XRP	-0.0019	0.44	1	0.62	0.71	0.68	0.64
XMR	0.54	0.82	0.62	1	0.89	0.91	0.86
ETH	0.44	0.91	0.71	0.89	1	0.95	0.83
ZEC	0.36	0.82	0.68	0.91	0.95	1	0.91
BCH	0.13	0.65	0.64	0.86	0.83	0.91	1

Figure 12 Pearson correlation heatmap of the analyzed 7 cryptocurrencies. December 2017

Now we get a different story altogether. In December 2017. Ether lost its tight correlation with BTC.

We then thought of, what if BTC/ETH correlation varies over time? Can we model it as a timeseries?

We can see the strong seasonality in the figure below. The daily correlation (not shown) is heavily oscillating. Since the seasonality moves to the monthly correlation, this ruled out what we thought that it was only due to some lag effect. (BTC moves, some time lag takes place, then ETH moves at the same direction). In fact, there were two instances where there was almost zero correlation between BTC and ETH. Those were in August (a hard fork time for BTC (Rizzo, 2017)) and in January (BTC bust)

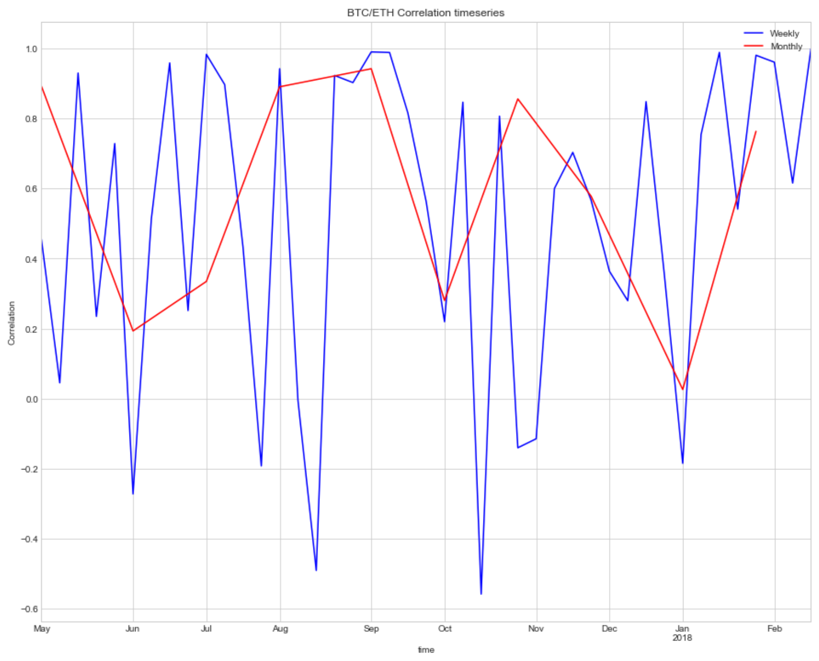


Figure 13 BTC/ETH weekly and monthly correlation. Since May 2017

We then plotted the daily and weekly autocorrelation[5]. Time dependence is shown on the daily autocorrelation and then disappears on the weekly. Time dependence is confirmed if one of the below vertical lines (showing correlation between the data and its lagged version) falls above/below the blue confidence intervals.

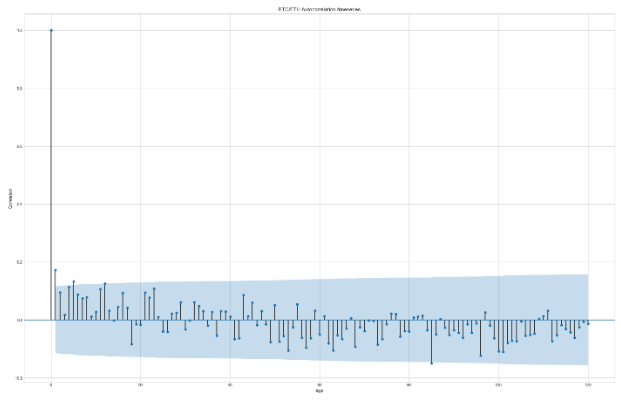


Figure 14 Daily autocorrelation. BTC/ETH

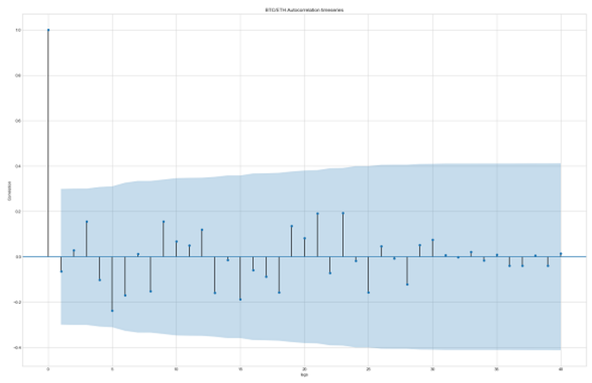


Figure 15 Weekly autocorrelation. BTC/ETH

Applying Machine Learning: Bitcoin Prediction

We applied different regression models on Bitcoin and Ethereum data with R2 goodness of fit (Frost, 2013) up to 87%. The timeseries nature of the data have limited the accuracy of any regression model and we believe this accuracy was as much as we could get.

Random Forest, Gradient Boosting and Extra Trees have had pretty much an equivalent quality of fit results on both BTC and ETH. Whereas Bayes Ridge and Elastic Net CV (Pedregosa, 2011) had much lower accuracy. We did not expect Elastic Net to have a chance as it applies cross validation on a timeseries data.

We have then tested the ExtraTrees on predicting 30 days of unseen data of Bitcoin and Ethereum. Table below shows the different R2 and training errors of each model.

	Bitcoin (BTC)	Ethereum (ETH)
Random Forest Regressor	R2: 0.86	R2: 0.84
	MAE: 988.69	MAE: 67.26
	MSE: 2369090.13	MSE: 8806.31
Gradient Boosting Regressor	R2: 0.86	R2: 0.81
	MAE: 1018.09	MAE: 71.01
	MSE: 2459920.09	MSE: 10267.14
ExtraTrees Regressor	R2: 0.87	R2: 0.82
	MAE: 910.91	MAE: 69.51

[5] More info: Time Series Basics. Penn State University. https://onlinecourses.science.psu.edu/stat510/?q=book/export/html/41

	MSE: 2263151.37	MSE: 10139.38
Bayesian Ridge	R2: 0.70	R2: 0.47
	MAE: 1714.85	MAE: 106.94
	MSE: 5178364.12	MSE: 29131.39
Elastic Net CV	R2: 0.66	R2: 0.35
	MAE: 1831.29	MAE: 143.78
	MSE: 5841023.25	MSE: 36031.04

Table 1 Regression output of Bitcoin and Ethereum daily average prices

Surprisingly the model generalized better with Bitcoin. It predicted a relatively similar price dip. Whereas it failed to predict Ether price with a relatively equivalent accuracy compared with the training data.

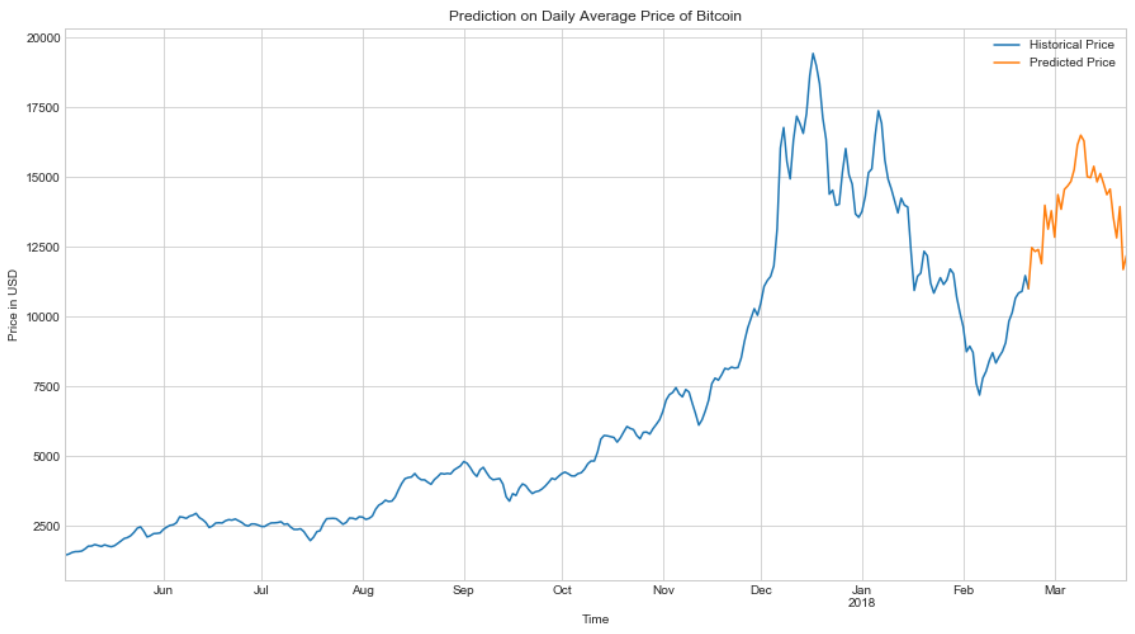


Figure 16 Bitcoin Prediction. 30 days from 22/2/2018

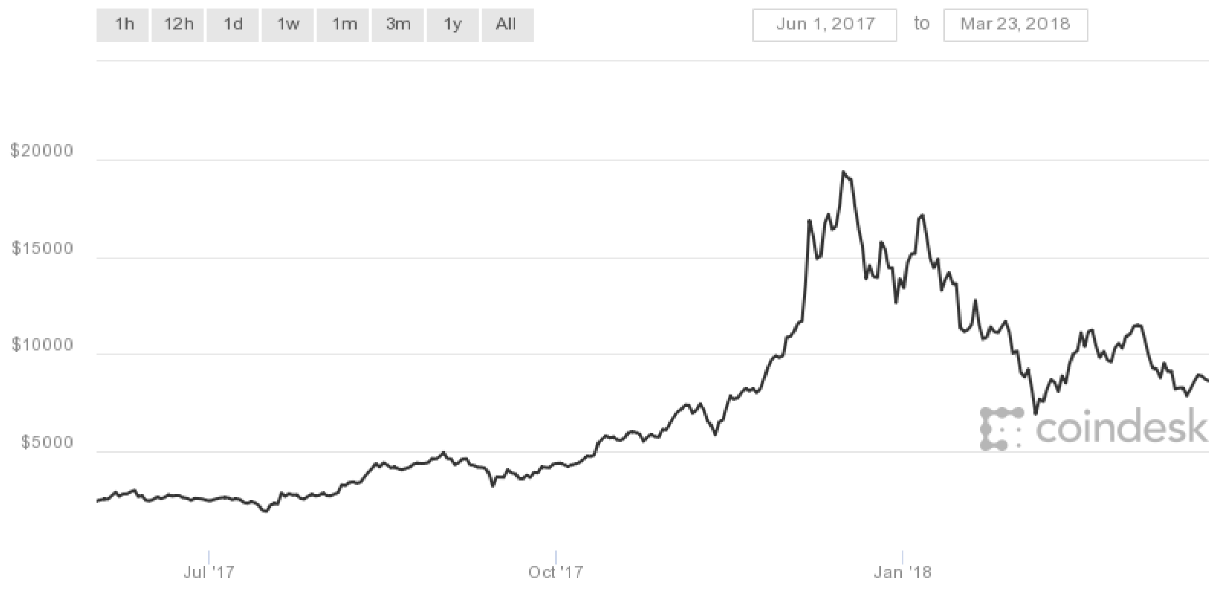


Figure 17 Coindesk actual BTC price chart. Up to 23/3/2018

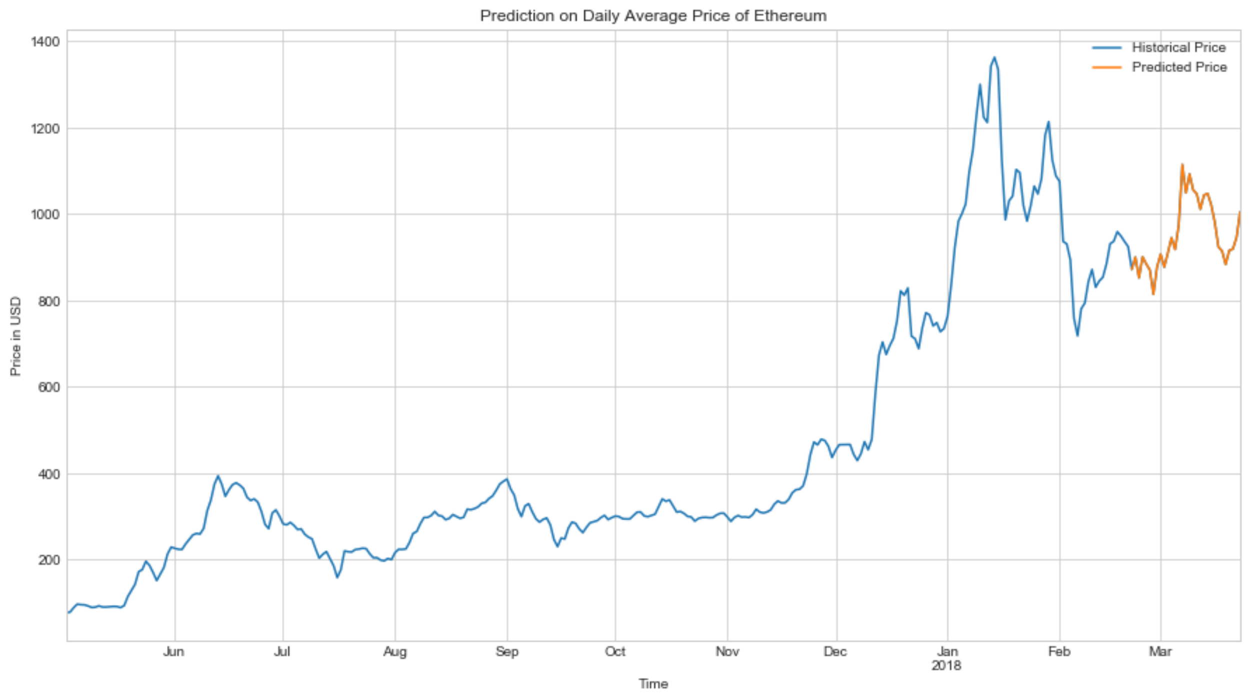


Figure 18 Ethereum Prediction. 30 days from 22/2/2018

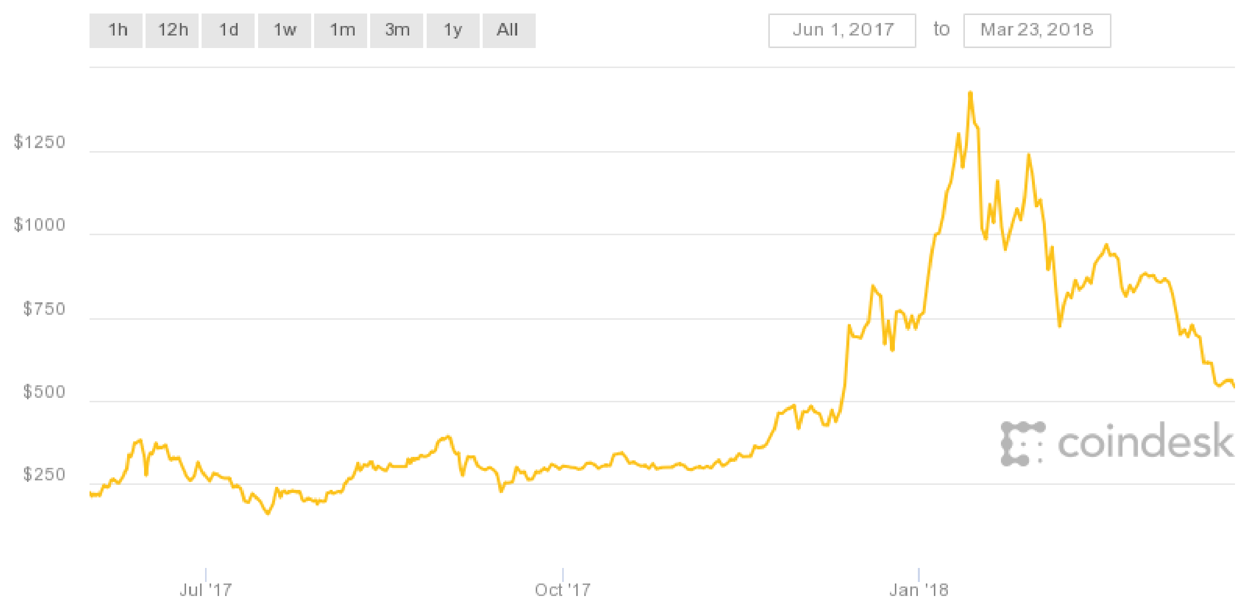


Figure 19 Coindesk actual ETH price chart. Up to 23/3/2018

Conclusion:

Cryptocurrencies are here to stay. We know that and exploring it in depth was really insightful.

We learned a lot of concepts about how the blockchain works. What are the challenges of data wrangling and how it becomes painful at times to extract and transform data.

We analyzed Ethereum smart contracts and found how huge their market share and potential is. They overpassed regular Ether transaction volume since mid-2017. Taking a closer look at the top smart contracts, we found that major leaders are cryptocurrency exchanges which raises a few questions of their own. For furthering our analysis, we believe a closer look into each smart contract transaction and looking at the statistical distribution these transactions would reveal some interesting insights.

We then dived into how Ethereum and Bitcoin are correlated. Before assessing this correlation, we explored how Ethereum market share is comparable to Bitcoin. We surprisingly found a short period where Ether transaction volume surpassed Bitcoin. Bitcoin was the most volatile cryptocurrency and we have seen that on a smoothed daily average plot. Ether has the second market cap after BTC. However, it was not the most volatile currency we explored. Yet it still generally follows a similar trend to Bitcoin.

We found two occasions when Ether and BTC had zero correlation. And we showed how this correlation strongly and predictably varies over time.

We finally managed to predict the price of Bitcoin with some success. And explored the timeseries nature of the dataset.

Future Work

We attempted to implement the ARIMA model based on (Brownlee, 2017). However, we needed more work to understand how to use it in prediction. The code can be found in the Jupyter notebook (See Appendix A).

We would also be very interested to apply this project on hourly data. We showed that the time series nature of the correlation data disappears if we have a long enough period.

Our Ethereum crawler needs some tuning as well and this would lead to a deeper smart contract analysis.

Individual Contribution

95% of the work is done by Mohamed Baddar and Muhammad Omar Waqar.

Daniyal Asad Chughtai have contributed in a few pages on the report. Ahmed had contributed on the final day of the coursework.

Appendix A: Crawler Code Overview

Our code is hosted on the following github repository:

https://github.com/mbaddar/Ethereum_Blockchain_Parser_Parallel

Below is a list of the most relevant files:

Scripts/parseBlockchain.py

This is the entry point to call the Ethereum blockchain crawler. The crawling code is based on (Miller, 2017) repository (https://github.com/alex-miller-0/Ethereum_Blockchain_Parser)

Preprocessing/Crawler/

This is the crawler class called by the parser above. It sends JSON-RPC requests to a geth node and inserts the response into a mongo database. It first checks the highest block available on the database then iterates from the next block to the maximum block specified

We set `min_block_geth = 2900000` that was mined on 29/12/2016 10:38 PM

The below is the crawler loop:

```python
# Get all new blocks
if self.max_block_mongo > self.min_block_geth and self.max_block_mongo < self.max_block
_geth:
    #Crawl from the next block
    print("Resuming crawling from block %8d" % ( self.max_block_mongo+1  ) )
    self.min_block_geth = self.max_block_mongo+1
try:
    self.min_block_geth = self.max_block_mongo
except Exception:
    print("Error …")
print("Processing the specified range of the blockchain from %8d to %8d..." % (self.min
_block_geth, self.max_block_geth))
for n in tqdm.tqdm(range(self.min_block_geth, self.max_block_geth)):
    self.add_block(hex(n))
print("Done!\n")
```

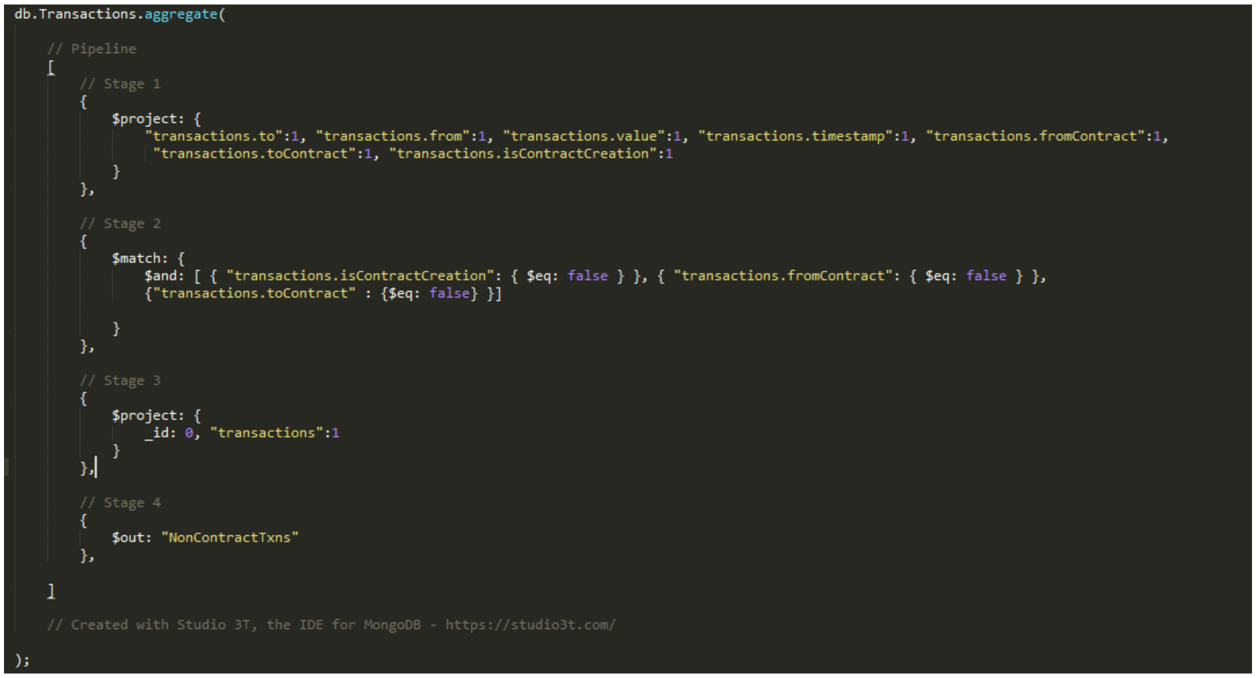
Appendix B: Data Extraction & Smart Contract Analysis Code Overview

Below is a list of the most relevant files:

Data Extraction and Exploration/Analysis

This folder contains the MongoDB aggregation queries, each numbered by a step number to be run in that order one by one. These queries result in creating additional collections that are either consumed by subsequent queries or the visualization code directly.

As an example, one of the queries is explained below:

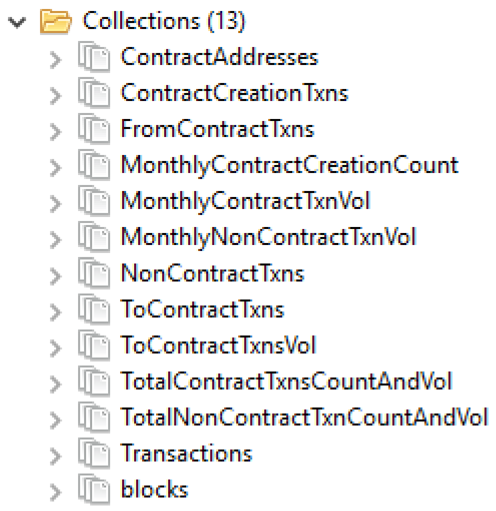


```
db.Transactions.aggregate(

    // Pipeline
    [
        // Stage 1
        {
            $project: {
                "transactions.to":1, "transactions.from":1, "transactions.value":1, "transactions.timestamp":1, "transactions.fromContract":1,
                 "transactions.toContract":1, "transactions.isContractCreation":1
            }
        },

        // Stage 2
        {
            $match: {
                $and: [ { "transactions.isContractCreation": { $eq: false } }, { "transactions.fromContract": { $eq: false } },
                {"transactions.toContract" : {$eq: false} }]

            }
        },

        // Stage 3
        {
            $project: {
                _id: 0, "transactions":1
            }
        },

        // Stage 4
        {
            $out: "NonContractTxns"
        },

    ]

    // Created with Studio 3T, the IDE for MongoDB - https://studio3t.com/

);
```

The aggregation pipeline works in stages where the output of last stage is the output of the next.

In the code above, first stage extracts only the relevant data fields so we don't have to pass unnecessary data in to later stages that would affect memory use. The next stage extracts the documents that match our given criteria. The third stage extracts only the transaction data to be saved in a collection by the name of "NonContractTxns" in the fourth stage.

After running all the queries, the list of collections looks like this:



```
Collections (13)
  ContractAddresses
  ContractCreationTxns
  FromContractTxns
  MonthlyContractCreationCount
  MonthlyContractTxnVol
  MonthlyNonContractTxnVol
  NonContractTxns
  ToContractTxns
  ToContractTxnsVol
  TotalContractTxnsCountAndVol
  TotalNonContractTxnCountAndVol
  Transactions
  blocks
```

The original data had only 'blocks' collection.

The visualization code is written in python. The 'MainScript.py' connects to MongoDB and uses

'VisualizationUtility.py' to plot different graphs and save them in '.png' format in the 'Results' folder.

For further information about the code, refer to the actual python code file.

Appendix C: Analytics Jupyter Notebook Code Overview

Dataset

The (Vent, 2018) Kaggle dataset can be downloaded from here:
https://www.kaggle.com/jessevent/all-crypto-currencies/downloads/all-crypto-currencies.zip/13

Notebook Location

Analysis Notebook/Analysis_of_Cryptocurrency_Investments.ipynb

This is the Jupyter notebook that contains the correlation and time series analytics we have done. It is based on (Ma, 2018) repository
https://github.com/jieyima/Cryptocurrency_Investment_Analysis_and_Modeling

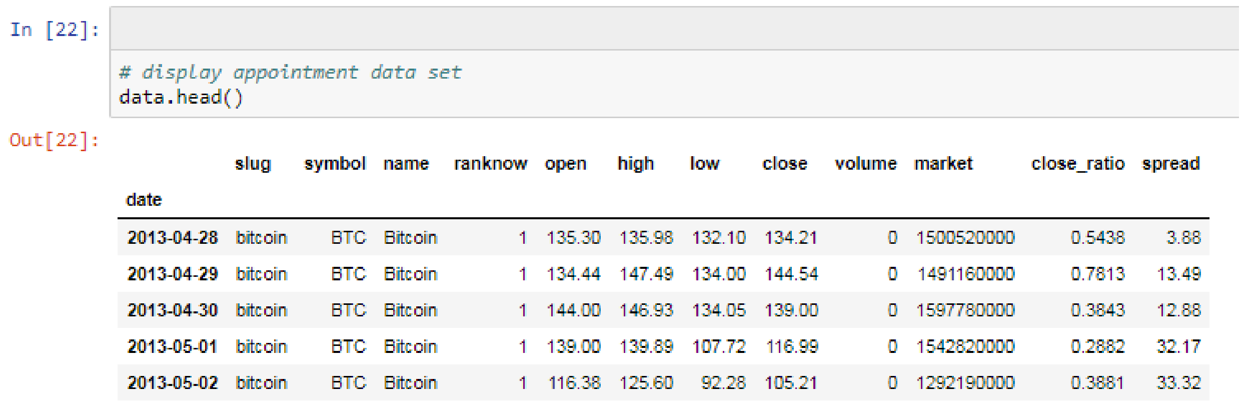
We have extended the correlation concept and attempted ARIMA analysis on bitcoin data.

Our contribution starts from plotting a correlation heatmap of a specific time period, then creating a correlation time series between bitcoin and Ethereum, then finally attempting to extend the regression models provided and apply ARIMA analysis on Bitcoin data.

Loading the dataset

```
data = pd.read_csv('crypto-markets.csv', parse_dates = ['date'], index_col = 'date')
```

The dataframe indexes the data by date. Here is a sample:



```
In [22]:
# display appointment data set
data.head()
```

Out[22]:

date	slug	symbol	name	ranknow	open	high	low	close	volume	market	close_ratio	spread
2013-04-28	bitcoin	BTC	Bitcoin	1	135.30	135.98	132.10	134.21	0	1500520000	0.5438	3.88
2013-04-29	bitcoin	BTC	Bitcoin	1	134.44	147.49	134.00	144.54	0	1491160000	0.7813	13.49
2013-04-30	bitcoin	BTC	Bitcoin	1	144.00	146.93	134.05	139.00	0	1597780000	0.3843	12.88
2013-05-01	bitcoin	BTC	Bitcoin	1	139.00	139.89	107.72	116.99	0	1542820000	0.2882	32.17
2013-05-02	bitcoin	BTC	Bitcoin	1	116.38	125.60	92.28	105.21	0	1292190000	0.3881	33.32

Figure 20 A sample of the crypto daily prices dataset

Correlation heatmap of a specific time period

```
 import datetime as dt

plt.figure(figsize=(12,6))
sns.heatmap(close['30-11-2017':'01-01-2018'].corr(),vmin=0, vmax=1, cmap='coolwarm', annot=True)
plt.title('Correlation Heatmap between Bitcoin and other 7 Cryptos')
plt.show()
```

Periodic Correlation

This function models periodic correlation with a time interval timedelta:

```
def calculate_corr(data, date,timedelta, column_name ):
    #Weekly correlation
    corr = []
    date_index = []
    while date <  dt.datetime(2018,2,21,0,0,0):
```

```python
        date_index.append(date)
        correlation = close[date :date+timedelta][['BTC','ETH']].corr()
        date = date+timedelta
        corr.append(correlation['BTC']['ETH'])
    return pd.DataFrame(corr, index=date_index, columns = [column_name])
```

It returns a dataframe with a periodic correlation from a start date with timedelta increments.

Regression Preprocessing

We shift the data 30 days and split the data into test and training sets

```python
BTC_before = data[data.symbol == 'BTC'].copy()
BTC['daily_avg_After_Month']=BTC['daily_avg'].shift(-30)
X_BTC = BTC.dropna().drop(['daily_avg_After_Month','symbol','daily_avg'], axis=1)
y_BTC = BTC.dropna()['daily_avg_After_Month']
X_train_BTC, X_test_BTC, y_train_BTC, y_test_BTC = train_test_split(X_BTC, y_BTC, test_
size=0.2, random_state=43)
X_forecast_BTC =  BTC.tail(30).drop(['daily_avg_After_Month','symbol','daily_avg'], axi
s=1)
```

Regression Models

This function applies the five regressors we used:

```python
# define regression function
def regression(X_train, X_test, y_train, y_test):
    Regressor = {
        'Random Forest Regressor': RandomForestRegressor(n_estimators=200),
        'Gradient Boosting Regressor': GradientBoostingRegressor(n_estimators=500),
        'ExtraTrees Regressor': ExtraTreesRegressor(n_estimators=500, min_samples_split
=5),
        'Bayesian Ridge': BayesianRidge(),
        'Elastic Net CV': ElasticNetCV()
    }

    for name, clf in Regressor.items():
        print(name)
        clf.fit(X_train, y_train)
```

Price Prediction

We call this function for price prediction.

```python
def prediction(name, X, y, X_forecast):
    model = ExtraTreesRegressor(n_estimators=500, min_samples_split=5)
    model.fit(X, y)
    target = model.predict(X_forecast)
    return target
```

Then add 30 days of prediction data to our dataset before plotting it:

```python
forecasted_BTC = prediction('BTC', X_BTC, y_BTC, X_forecast_BTC)
# define index for next 30 days
last_date=data.iloc[-1].name
modified_date = last_date + dt.timedelta(days=1)
```

```
new_date = pd.date_range(modified_date,periods=30,freq='D')

# assign prediction to newly defined index
forecasted_BTC = pd.DataFrame(forecasted_BTC, columns=['daily_avg'], index=new_date)
```

Python Modules Used

On top of pandas, matplotlib, and scikit-learn. We used the following Python modules:

- **Pymongo:** To interface with mongodb
- **Requests**: an HTTP library to send get requests to the Geth node
- **Tqdm**: make loops show a smart progress meter
- **Statsmodels:** provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests.

Appendix D: Output overview

An example transaction extracted from the blockchain in json format:

```
"transactions": [{
        "fromContract": false,
        "gasPrice": NumberLong(20000000000),
        "gas": NumberInt(134470),
        "txNumber": NumberInt(1),
        "to": "0xe94b04a0fed112f3664e45adb2b8915693dd5ff3",
        "from": "0x2a73584d7a36f345524b5c74d9ee2a8f8d9ef158",
        "timestamp": NumberInt(1483051090),
        "toContract": false,
        "isContractCreation": false,
        "inputData": "0x0f2c9329000000000000000000000000fbb1b73c4f0bda4f67dca266ce6ef42f520fbb
98000000000000000000000000e592b0d8baa2cb677034389b76a71b0d1823e0d1",
        "number": NumberInt(2900000),
        "value": 0.11236934
    },
… More transactions …
```

Running a full Ethereum node taking a few days to synchronize:

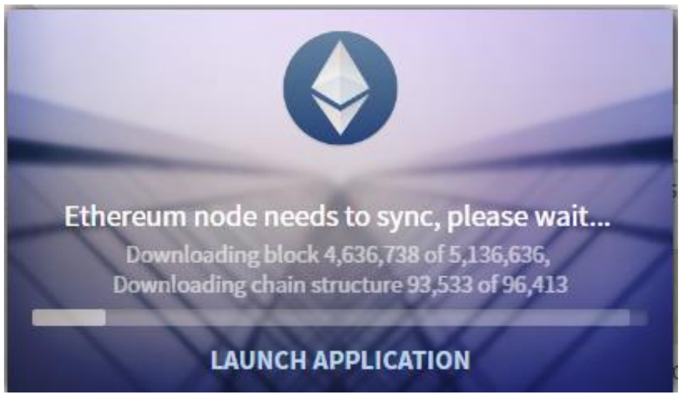
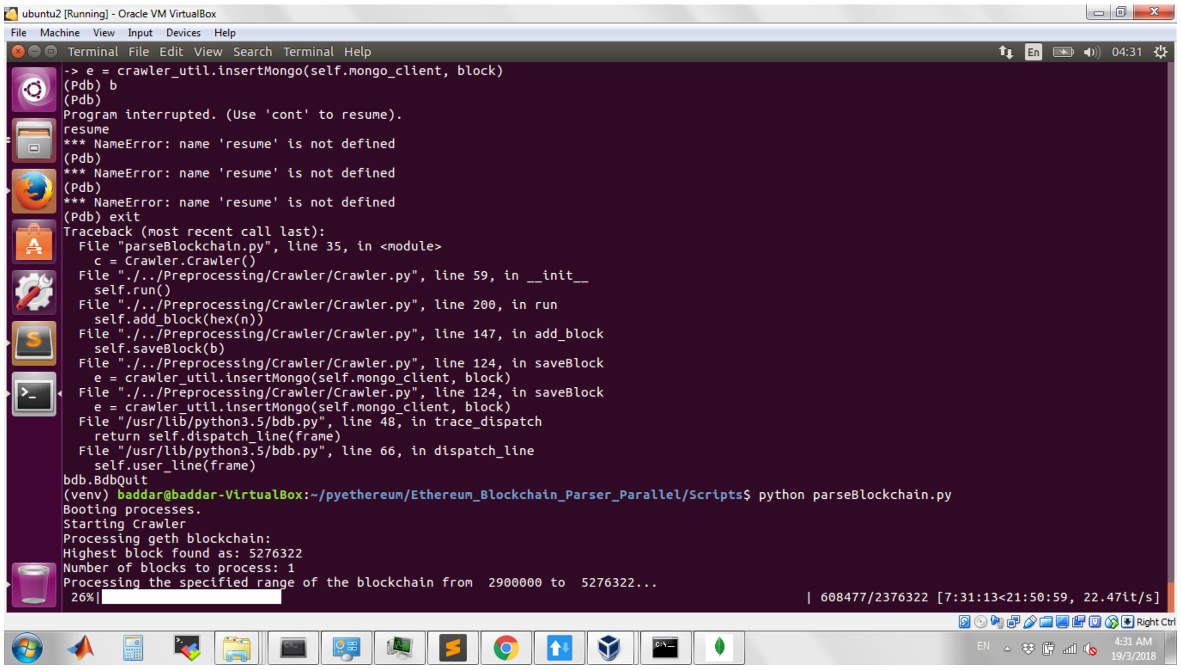

Ethereum node needs to sync, please wait...
Downloading block 4,636,738 of 5,136,636,
Downloading chain structure 93,533 of 96,413

Figure 21 Synching a full Ethereum node. A new block gets generated every 15 seconds

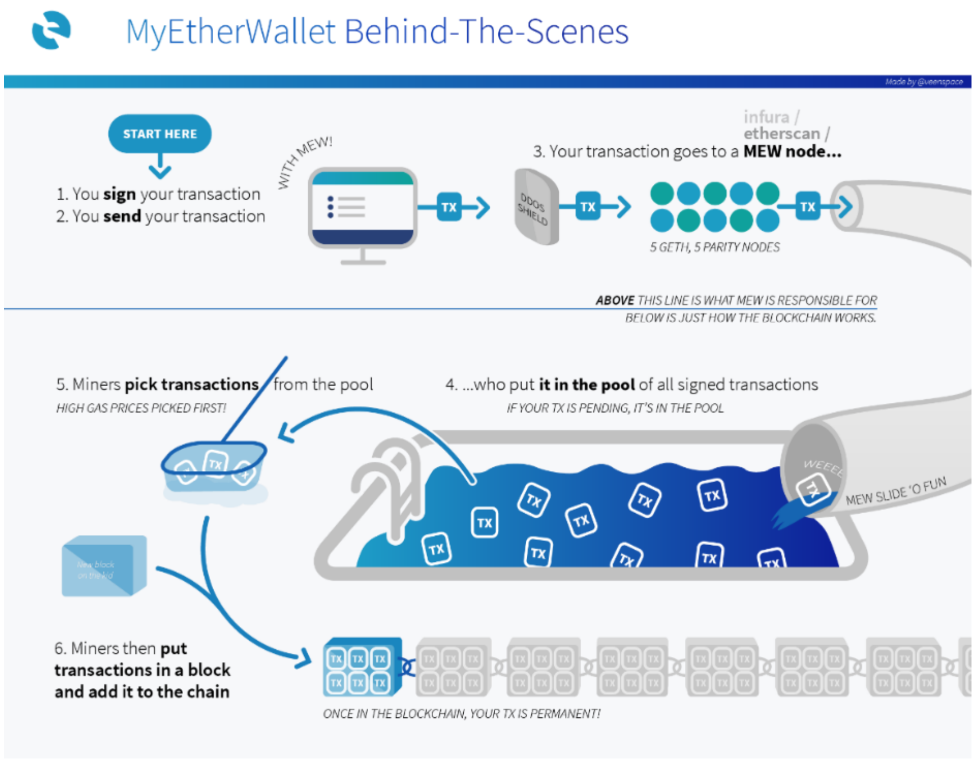
Our crawling code running on a Virtual Box Linux machine:


ubuntu2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal File Edit View Search Terminal Help
04:31
Right Ctrl
4:31 AM
19/3/2018

```
-> e = crawler_util.insertMongo(self.mongo_client, block)
(Pdb) b
(Pdb)
Program interrupted. (Use 'cont' to resume).
resume
*** NameError: name 'resume' is not defined
(Pdb)
*** NameError: name 'resume' is not defined
(Pdb)
*** NameError: name 'resume' is not defined
(Pdb) exit
Traceback (most recent call last):
  File "parseBlockchain.py", line 35, in <module>
    c = Crawler.Crawler()
  File "./../Preprocessing/Crawler/Crawler.py", line 59, in __init__
    self.run()
  File "./../Preprocessing/Crawler/Crawler.py", line 200, in run
    self.add_block(hex(n))
  File "./../Preprocessing/Crawler/Crawler.py", line 147, in add_block
    self.saveBlock(b)
  File "./../Preprocessing/Crawler/Crawler.py", line 124, in saveBlock
    e = crawler_util.insertMongo(self.mongo_client, block)
  File "./../Preprocessing/Crawler/Crawler.py", line 124, in saveBlock
    e = crawler_util.insertMongo(self.mongo_client, block)
  File "/usr/lib/python3.5/bdb.py", line 48, in trace_dispatch
    return self.dispatch_line(frame)
  File "/usr/lib/python3.5/bdb.py", line 66, in dispatch_line
    self.user_line(frame)
bdb.BdbQuit
(venv) baddar@baddar-VirtualBox:~/pyethereum/Ethereum_Blockchain_Parser_Parallel/Scripts$ python parseBlockchain.py
Booting processes.
Starting Crawler
Processing geth blockchain:
Highest block found as: 5276322
Number of blocks to process: 1
Processing the specified range of the blockchain from  2900000 to  5276322...
 26%|                                                                 | 608477/2376322 [7:31:13<21:50:59, 22.47it/s]
```

Figure 22 Crawling the blockchain. Early blocks had a bigger throughput (22.7 block/sec) as they contained smaller number of transactions. It then went down to about 6 blocks/sec

A typical Ether transaction flow:



MyEtherWallet Behind-The-Scenes
Made by @veenspace
START HERE
1. You sign your transaction
2. You send your transaction
WITH MEW!
TX
DDOS SHIELD
infura / etherscan /
3. Your transaction goes to a MEW node...
5 GETH, 5 PARITY NODES
ABOVE THIS LINE IS WHAT MEW IS RESPONSIBLE FOR
BELOW IS JUST HOW THE BLOCKCHAIN WORKS.
5. Miners pick transactions from the pool
HIGH GAS PRICES PICKED FIRST!
4. ...who put it in the pool of all signed transactions
IF YOUR TX IS PENDING, IT'S IN THE POOL
WEEE!
MEW SLIDE 'O FUN
New block on the kid
6. Miners then put transactions in a block and add it to the chain
ONCE IN THE BLOCKCHAIN, YOUR TX IS PERMANENT!

Figure 23 A typical Ethereum transaction flow. MyEtherWallet

Bibliography

About VirtualBox. (2018). Retrieved from virtualbox.org: https://www.virtualbox.org/wiki/VirtualBox

AminCad. (2018, 1 12). *Market share of Ethereum-based tokens grows to 91%*. Retrieved from Medium: https://medium.com/@amincad/market-share-of-ethereum-based-tokens-grows-to-91-fdefadfd9f6e

bitcoin.org. (2018). *Bitcoin Developer Guide*. Retrieved from bitcoin.org: https://bitcoin.org/en/developer-guide#block-chain

Brownlee, J. (2017). *How to Create an ARIMA Model for Time Series Forecasting with Python*. Retrieved from machinelearningmastery.com: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

Buterin, V. (2014). *Ethereum Scalability and Decentralization Updates*. Retrieved from Ethereum Blog: https://blog.ethereum.org/2014/02/18/ethereum-scalability-and-decentralization-updates/

Castillo, E. (2018). *The Ethereum White Paper*. Retrieved from The Ethereum Wiki: https://github.com/ethereum/wiki/wiki/White-Paper#computation-and-turing-completeness

Ethereum.org. (2018). *Go Ethereum. Official Go implementation of the Ethereum protocol*. Retrieved from geth.ethereum.org: https://geth.ethereum.org/

Ethereum.org. (2018). *JSON-RPC*. Retrieved from github.com: https://github.com/ethereum/wiki/wiki/JSON-RPC

Frost, J. (2013, 5 30). *Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?* Retrieved from minitab.com: http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit

Ghemawat, S., & Dean, J. (n.d.). *ethereum/leveldb*. Retrieved from Github: https://github.com/ethereum/leveldb

Griffith, K. (2014, 4 16). *A Quick History of Cryptocurrencies BBTC — Before Bitcoin*. Retrieved from Bitcoin Magazine: https://bitcoinmagazine.com/articles/quick-history-cryptocurrencies-bbtc-bitcoin-1397682630/

Investopedia. (n.d.). *Initial Coin Offering (ICO)*. Retrieved from Investopedia: https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp

Investopedia. (n.d.). *Smart Contracts*. Retrieved from Investopedia.com: https://www.investopedia.com/terms/s/smart-contracts.asp

Liu, B. Y. (2018). *Ethereum Development Tutorial*. Retrieved from github.com: https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial

Ma, M. (2018, 3). *Cryptocurrency_Investment_Analysis_and_Modeling*. Retrieved from github.com: https://github.com/jieyima/Cryptocurrency_Investment_Analysis_and_Modeling

Madeira, A. (2018, 4 5). *The DAO, The Hack, The Soft Fork and The Hard Fork*. Retrieved from Cryptocompare.com: https://www.cryptocompare.com/coins/guides/the-dao-the-hack-the-soft-fork-and-the-hard-fork/

Miller, A. (2017). *Ethereum_Blockchain_Parser*. Retrieved from github.com: https://github.com/alex-miller-0/Ethereum_Blockchain_Parser

MyEtherWallet. (2018). *What is Gas?* Retrieved from myetherwallet: https://myetherwallet.github.io/knowledge-base/gas/what-is-gas-ethereum.html

Nakamoto, S. (2008, 10). Retrieved from Bitcoin.org: https://bitcoin.org/bitcoin.pdf

Pedregosa. (2018). *http://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting*. Retrieved from Sciki-learn.org: http://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting

Pedregosa, F. a. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2825--2830.

Ray, J. (2018, 4). *The Ethereum White Paper*. Retrieved from The Ethereum Wiki: https://github.com/ethereum/wiki/wiki/White-Paper#ethereum

Rizzo, P. (2017, 8). *Block 494,784: Segwit2x Developers Set Date for Bitcoin Hard Fork*. Retrieved from coindesk.com: https://www.coindesk.com/block-494784-segwit2x-developers-set-date-bitcoin-hard-fork/

Siebel, J. (2017, 7 21). *Ethereum ERC20 Tokens Explained*. Retrieved from Medium: https://medium.com/@james_3093/ethereum-erc20-tokens-explained-9f7f304055df

Sifr Data. (2018, 4). *Cryptocurrency Correlation Matrix*. Retrieved from Sifr Data: https://www.sifrdata.com/cryptocurrency-correlation-matrix/

Stack Exchange, U. (2017, 7 13). *Ethereum block architecture*. Retrieved from Stack Exchange: https://ethereum.stackexchange.com/questions/268/ethereum-block-architecture

Szabo, N. (2008, 12 29). *Bit Gold*. Retrieved from Nakamoto Institute: http://nakamotoinstitute.org/bit-gold/

The Ethereum Foundation. (2018). *Solidity*. Retrieved from github.com: https://github.com/ethereum/solidity/blob/v0.4.21/docs/index.rst

Throwback Thursday: PayPal's Biggest Days In History. (2015, 7 2). Retrieved from pymnts.com: https://www.pymnts.com/in-depth/2015/throwback-thursday-paypals-biggest-days-in-history/

Vent, J. (2018). *Every Cryptocurrency Daily Market Price*. Retrieved from Kaggle.com: https://www.kaggle.com/jessevent/all-crypto-currencies

Wikipedia. (n.d.). *Blockchain*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Blockchain

Wikipedia. (n.d.). *Cryptocurrency*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Cryptocurrency

Wikipedia. (n.d.). *David Chaum*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/David_Chaum

Wikipedia. (n.d.). *DigiCash*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/DigiCash

Wikipedia. (n.d.). *Double Spending*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Double-spending

Wikipedia. (n.d.). *Ethereum*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Ethereum

Wikipedia. (n.d.). *Nick Szabo*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Nick_Szabo

Wikipedia. (n.d.). *The DAO (organization)*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/The_DAO_(organization)

Williams-Grut, O. (2018, 1 31). *Only 48% of ICOs were successful last year — but startups still managed to raise $5.6 billion*. Retrieved from Business Insider: http://uk.businessinsider.com/how-much-raised-icos-2017-tokendata-2017-2018-1