Visualizing and Forecasting the Cryptocurrency Ecosystem

Shawn Anderson, Ka Hang Jacky Lok, Vijayavimohitha Sridhar

1 Motivation and Background

Cryptocurrencies are programmable digital assets which can be traded on an open market. The first cryptocurrency, Bitcoin, created in 2009, has a market capital of more then 140 billion dollars. In the past 10 years, the price of Bitcoin has grown 80,000 times. No one could have imagined this upsurge in cryptocurrency value. Some news has defined cryptocurrency as a fraud or bubble while others think that it can revolutionize the way humans interact with money. Popular opinion of Bitcoin is rich with mysticism. Everyone in the world is amazed by the surging high price of Bitcoin. People such as students, hackers, investors, banks, and government are all staring it and trying to make something out of it.

Related Work

In Evolutionary dynamics of the cryptocurrency market, EIBahrawy Et al. explore the birth and death rate of new cryptocurrencies, comparing them to that of animals in biological ecosystems. They find significant similarity in the mathematical modeling of biological ecosystems and cryptocurrency ecosystems.[1] In Deep Reinforcement Learning for Financial Portfolio Management, Jiang et. al. achieve highly rewarding results in implementing a reinforcement learning agent to manage a cryptocurrency portfolio in historical backtests. They pioneer the EIIE network topology to feed feed coin evaluations to a reinforcement learning agent which manages a portfolio. They are able to increase their principle investment 47X relative to the price of bitcoin in 50 days in one of their backtests [2]. In Predicting Cryptocurrency Prices With Deep Learning, Sheehan uses LSTM to model price movement of Bitcoin and Ethereum[3]. In Analyzing Cryptocurrency Markets Using Python, Triest does statistical analysis of a handful of top cryptocurrencies[4].

2 Problem Statement

As we began to brainstorm this project, we formulated three questions to investigate:

1. Can we visualize the cryptocurrency ecosystem?
2. Can we identify factors which effect the value of cryptocurrencies and the opinions of market participants?
3. Is it possible to predict the future value of cryptocurrencies based on past data?

These questions are challenging because they are completely self defined by our group. We had to do everything from scratch: background research on the subject, finding data sources, discovering how to collect the data we need. That was only the beginning, once we started identifying data sources, and implementing scraper and collection modules, we had to begin considering how to integrate the data and use it to answer our questions. These questions are rather profound, even for an expert of the domain. For example, if one is able to successfully answer question 2 or 3, then they are likely able to make a healthy profit off of their information. As for question 1, it is not as complex as the others, but demands a lot of creativity, as there is no definition of what should be visualized, we had to spend a lot of time trying out random things until we began to find something that makes sense.

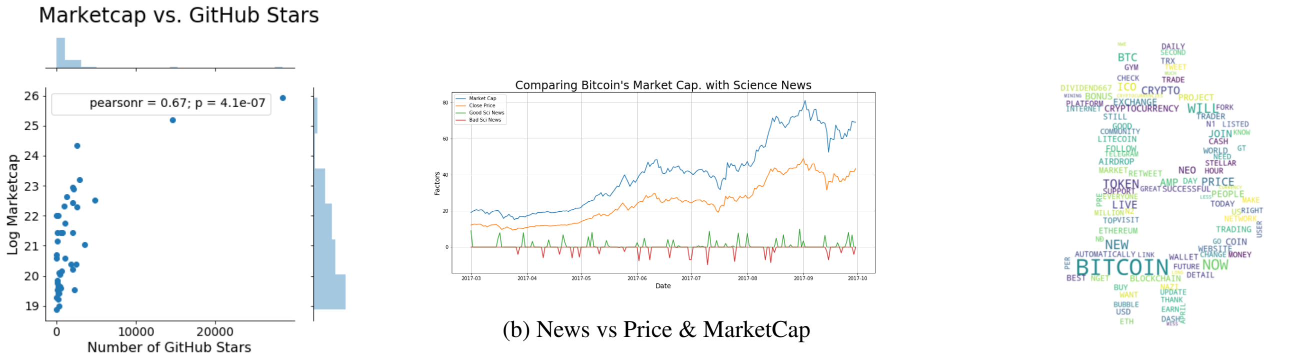
3 Data Science Pipeline

3.1 Data collection

We collected our data from coinmarketcap.com, Github, Wikipedia, and Twitter.

Coinmarketcap

Coinmarketcap.com is a website that lists hundreds of cryptocurrencies, ranked in order of their respective market cap sizes. We implemented a python module for automatically scraping these data points for all 100 coins listed on the front page of



(a) Github vs MarketCap

(b) News vs Price & MarketCap

(c) WordCloud from Tweets

coinmarketcap. We use price history data collected in this way for visualization and price forecasting. Links to external sources are collected for further scraping, in particular, we do this with links to GitHub URLs.

GitHub

GitHub is a website which serves as a social platform for hosting, sharing, cloning, forking, and collaborating on source code. All cryptocurrencies are open source, and thus their code bases can be found hosted on GitHub. We implement a GitHub scraper module that will return number of stars, forks, and watches given a repository URL. Our coinmarketcap scraper module returns a GitHub url for each coin that it scrapes. Together, these two modules are able to collect GitHub feature data for each of our coins. We then integrate the GitHub features with coin price data to produce visualizations such as 1a.

Wikipedia Current Events

Wikipedia Current Events is a Wikipedia subpage that records only the biggest News in the world on a daily basis. We implemented a Wikipedia scraper module that collects news that contains dates, categories, headlines, and references. Once the scraper started, it will collect data from 1995 up to current, using python program called Scrapy. The scraper module is capable to constantly running to update the database and automatically save the data in csv format.

Twitter

Twitter is one of the most widely used social media website, it contains one of the most update-to-date News in the world. We implemented a Twitter scrapper module that used Twitter streaming API to download Tweets that mentioned cryptocurrencies.

3.2 Data Preparation

We then clean the downloaded data, back filling missing data, joining datasets, doing entity resolution, chopping and reshaping of multi-dimensional tensors such that they make sense to be fed to a deep neural network, separation of input and target data for our model, separation of train, validation, and test sets for training.

3.3 Data Integration and Exploration

The key to our data integration was that all of our datasets where based around a unified list of keywords. These keywords were the names of the top 100 cryptocurrencies, ranked by marketcap. All of our datasets, all of our web scraping, were based on their relation to these 100 key words. In the case of external data such as historical events from wikipedia, datasets are integrated based on a date column.

3.4 Data Analysis

Our primary methods of analysis were visual inspection and deep learning. We used many analysis techniques along the way such as entity resolution, language visualization, statistical sampling methods, statistical projection and model validation.

4 Methodology

Exploratory Analysis with Jupyter Notebooks

Jupyter notebooks provide clean, interactive environments for reproducible research. In order to maximize efficiency as a team, we were each assigned separate data sources to perform collection, cleaning, analysis, and exploration on. To ensure that the work that each member did was transparent and clear to the others, we would always produce a jupyter notebook to display the progress and discoveries that we made. This process proved to be extremely rewarding by making it easy to recall and reuse previous work. It also has the beneficial side effect of producing a portfolio of the work that we have done.

Price Forecasting with Deep Learning

Inspired by the work of Jian et. al. [2], we investigated the possibility predicting future coin values based on historical price data using deep learning. The intuition that this would work comes from the domain of technical analysis. Technical analysis is a field of finance in which future asset prices can be calculated as probabilistic functions of past asset prices. Over the past century, technical analysts have discovered many mathematical equations to express this relationship between past and future prices.

The wonderful part about deep learning is the concept of representation learning, that is, if there are latent features within the data that are composed of complex combinations of lower level features, then a deep architecture can first learn to detect such latent features, and then use these latent features to make high level classifications or regressions. An example of this is in computer vision, where a deep CNN will learn a hierarchy of latent features, beginning with edge, and colour detection. Later layers of the network will continue to learn more complex latent features such as wheel or face detection. Similarly, we postulate that a deep CNN could learn a hierarchy of latent features to predict price movement, ultimately, *rediscovering* what mathematical economists, technical analysts, and quantitative financiers have labored and toiled to discover over the past century.

Tools and Analysis Methods

We used lxml, beautiful soap, and scapy for data collection. We chose them because they are widely used(so we can find more resources), stable and easy to use. We constructed them into our first data gathering layer, they can be either called directly by a python API or a daemon that keep running in the background.

We used NLTK for sentimental analysis and word cloud. We chose NLTK since they have a pre-trained model that are ready to use. We used NLTK to analysis Twitter comments and make wordcloud base on it. We also applied NLTK to do sentimental analysis on News Headline, which generated a list of good news and bad news.

We used Keras to train our CNN and LSTM model. Best results were acheived with our CNN implementation which utilized three 1D convolutional layers, each with max pooling and dropout. Followed by two fully connected layers with dropout. Followed by fully connected activation layer. The activation can be swapped out for different types of forecasting, sigmoid for binary prediction, linear for price prediction, or softmax for portfolio recommendations.

5 Evaluation

EDA in News

We plotted the good and bad sciences News side-by-side with Bitcoin market cap and price, and we found that when there is good news, the price goes up and when there is bad news, the price goes down. And we explored the detail News headline and checked they actually make sense.

WordCloud in Twitter

We checked our WordCloud made by Twitter EDA, and they matched closely with what people expected. Words such as Bitcoin, Trading and Token had comes up in the font.

Projecting Future Prices Based on Historical Distributions

In order to visualize the nature of long term price trends of cryptocurrencies, we plotted a histogram of daily price change distributions for the two most dominant coins in the market, Bitcoin and Ethereum 2. Daily price change is calculated as close price divided by opening price for each day.

From these distributions we can see that Ethereum is much more volatile than Bitcoin. In fact, there is one day where Ethereum's price dropped more than 60%, and multiple days where Ethereum's price increased by more than 40%. From these distributions we find that the mean daily price changes for Bitcoin and Ethereum are a factor of 1.003, and 1.0079 respectively, with standard deviations of 0.005, and 0.008. From this we can conclude that given their respective histories, Etherium has been growing in value faster than Bitcoin, but with more volatility.

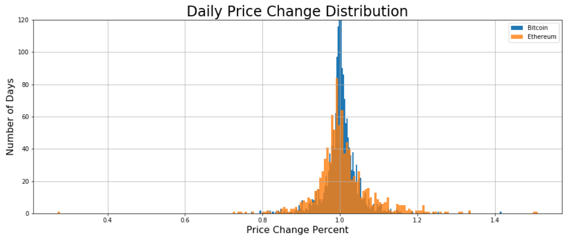


Figure 2: Distributions for of daily price changes for Bitcoin and Ethereum, the two most dominant cryptocurrencies. Daily price change is calculated as close price divided by opening price.

If we are to take these distributions as the true nature of growth with respect to time, then we can extrapolate future values to arbitrarily far into the future with the following basic compounding returns formula:

$$future\_price = current\_price * growth\_rate\_per\_day^{number\_days}$$

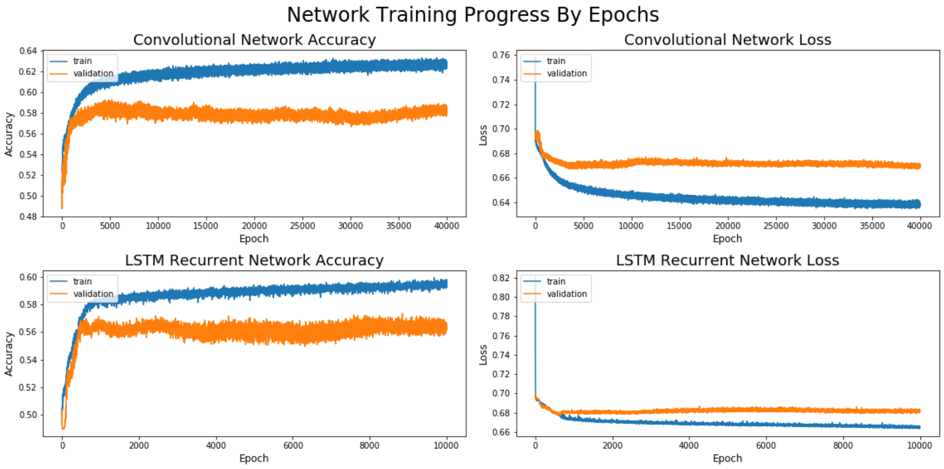
plugging price of these coins for April 15th 2018, the day of this writing, and extrapolating one year into the future we get the projected price predictions for Bitcoin and Ethereum:

$$\text{Projected Bitcoin price, April 15th 2019: } \$8289.51 * (1.003^{365}) = \$24738.27$$

$$\text{Projected Ether price, April 15th 2019: } \$523.90 * (1.0079^{365}) = \$9260.10$$

Price Prediction

We have got 58% and 56% validation accuracy with CNN and LSTM, respectively, for binary next day prediction. Although that may sound low, if you are correct on even 51% of bets in the long run, you can make a profit. The chance of getting 58% accuracy by random chance on 1396 validation examples is less than 1 in 100 million[7]. Also, you can see that accuracy improves, and loss decreases up until around the 5000 Epoch range, so clearly the model is learning how to predict future prices.



6 Data Product

Our data product is a collection of jupyter notes and python modules. We can categorize the product in three parts: scraper modules, a cryptocurrency analyzer, and a cryptocurrency forcaster.

CryptoViz Web Scraper Module

We have four web scrapers, each is crawling from different source and using different tools. For Coinmarketcap and github scraper, it can be easily used by importing the python package and call one line of command. See 8

When calling these functions, the program will automatically scrape data from coinmarketcap or github and convert it to pandas objects. Wikipedia News scraper is a bit different, we will run scrapy to activate it and it will download data in json file and save it to the specific folders. On the other hand, Twitter scraper, used streaming API to pull twitter data and download as json file. We can call the scrap python directly to start the scraping.

Cryptocurrency Analyzer

Cryptocurrency analyzer is an EDA process to visualize crytocurrencies. It gives you the general idea of crytocurrency, for example: the market capital of each coins, the relationship between market cap and Github starts and the relationship between price, market cap and news topics. The analyzer modules are saved in Jupyter Notebooks and is ready for anyone to follow instructions and run the analysis. We have three different EDA process in our project, one for coinMarketCap and github stars which used to analysis the movement of each coin's market cap overtime, and Github stars over market cap. In News EDA, we analysis the relationship between specific News topics versus Bitcoin's price and market cap. In Tweeter EDA, we found the tweets per hours and wordcloud about cryptocurrency.

Cryptocurrency Forcaster

Our cryptocurrency forecastor uses deep neural networks to ouput an evaluation of all 100 coins from our dataset. Currently we have implemented next day binary forecasting. This means that our model will assign a score between 0 and 1 to every coin. It will sort the coins in descending order and display them on the screen along with their ranking. Coins with scores closest to 1 should be bought, coins with scores closest to 0 should be sold. Our model is able to make recommendations for 10 and 30 days in the future as well. Currently switching between recommendation times requires retraining the model, in future work, we plan to use transfer learning to only retrain the final dense layers when switching between recommendation times.

Right now our model produces binary predictions, meaning that it is predicting whether a coin will go up or down in value. There are two other forecasting options that our model is capable of. Firstly, linear forecasting, in which the model ouputs the actual predicted price of the coins, this is a transition from a binary classification task to a regression task and only requires changing the output activation function of our model. Secondly, portfolio recommendations, in which the model outputs a softmax distribution over all coins. A softmax distribution is a vector that sums to one, with non-negative elements. These are exactly the properties of an investment portfolio. Currently, changing the forecasting option requires retraining the model, in future work we will use transfer learning to only retrain the final dense layers when changing the forecasting option.

7 Lessons Learned

We learned to think as data scientists - collecting our own data and performing statistical evaluations to find interesting results. In the data collection process, we tried multiple methods and tools for scraping data from different sources. We also managed to clean data, deal with missing data (Ex: back filling missing data), remove duplicate data using entity resolution, and reshaping data for later process. We learned how to integrate language processing (sentimental analysis) to empower the analytic process. We learned how to use powerpoint to make an awesome video, and how to use our passion to drive us to produce a data product that we can be proud of.

8 Summary

Data visualization has provided insights into the nature of cryptocurrencies and their relationship to News, GitHub and Twitter. We have shown Convolutional Networks to be a reliable architecture for cryptocurrency price forecasting. With additional data and tuning, we see a potential for use in production. The answers we found have lead to many more questions. We see cryptocurrencies as profoundly historical phenomenon that will impact society in many ways. We look forward to future work done on this subject.

References

[1] Anne Kandler Romualdo Pastor-Satorras Andrea Baronchelli Abeer ElBahrawy, Laura Alessandretti. Evolutionary dynamics of the cryptocurrency market, 2017.

[2] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. A deep reinforcement learning framework for the financial portfolio management problem, 2017.

[3] David Sheehan. predicting-cryptocurrency-prices-with-deep-learning. https://dashee87.github.io/deep%20learning/python/predicting-cryptocurrency-prices-with-deep-learning/.

[4] Patrick Triest. Analyzing cryptocurrency markets using python. https://blog.patricktriest.com/analyzing-cryptocurrencies-python/.

Appendix

```python
import numpy as np
trials = []
for i in range(10000000):
  trials.append(np.random.binomial(1396, 0.5))
threshold = 0.58*1396
[1 for t in trials if t >threshold]
```

Listing 1: Sampling code used to test likelihood of model validation results being acheived through randomness. It happens zero times in 100 Million samples

```python
from Scrapers.Coinmarketcap import coinmarketcap
# Example of using coinmarketcap scraper
cmk = coinmarketcap.CoinMarketcap()
coins = cmk.coins()
# Example of using github scraper
bitcoin = coins[0]
btc_repo = bitcoin.repo()
```

Listing 2: An example use of the CryptoViz scraper module.

Code Repository: https://github.com/LinuxIsCool/733Project
Project Poster: https://shawnwanderson.github.io/pdf/Final-733Poster.pdf