Final report : Cryptocurrency prediction using LSTM & feature selection

EE443/EE543, Northern Arizona University

Rogelio Cabrera, Javier Herrer, Félix Roux

Abstract

Our project is an attempt on predicting the Bitcoin exchange value and variation over time, with a focus on the analysis of new ways to aggregate data to improve prediction precision. As we found really early in our project, there is no existing algorithm able to accurately predict Bitcoin value over time. However, we try to propose new trails for methodologies that could resolve that problem in new and unseen ways.

We based our result validation solution on a LSTM Neural Network, with a custom hyperparameter optimization algorithm. This algorithm allowed us to feed it with rather brute data, facilitating the process of development.

Our goal with this implementation was to prove that only analysing the exchange value of Bitcoin was not enough to predict it’s value, despite a virtually really optimized algorithm. It can also serve as a framework for implementing other types of data, to explore new possibilities that will be developed later in the document.

Brief introduction

For the sake of simplicity, we will only focus on Bitcoin throughout the entire project, as it is the most well-known cryptocurrency in circulation as of now, and is one of the oldest : which means that we can access a lot of data quite easily.

One of the main reasons that make all “cryptocurrency value prediction” algorithms fail is the lack of data. There are two majors ways of “gaining” data to solve problems :

- Have more data length : older data, or more points of data per unit of time. As the exchange value of Bitcoin is updated on a regular basis, our data resolution would be capped : this is not the approach we will take.

- Have more data dimensions : more different data. We took this path, as our original intention was to try to implement other types of data for the algorithm to have more ways to identify patterns

Our original idea was to use that second method : as stock markets, and especially cryptocurrencies, are influenced by a huge number of factors, we wanted to introduce other features, such as Tweets analysis, to give the algorithm more insight over what influences and is influenced by certain text elements.

Main Section

Twitter analysis

Principle

For this project, we wanted to have an element that could make it stand out from the rest of the projects for the class, and that could make it also stand out from other crypto-related projects. As described earlier, we wanted to include a text analysis tool in our program, and, using the Twitter API, retrieve Tweets from the platform. That way, we could associate text elements with Bitcoin values, and thus allow our algorithm to better understand complex phenomenons.

This method is quite interesting, as it can be used in multiple ways :

- We can select which Tweets we would like to retrieve with precise filters : the Twitter API is precise enough to let us sort through Tweets according to Hashtags, key-words, users, *etc*…
- We can filter the text in many different ways : we can include the Emojis in the analysis, the hyperlinks, ignore capitalization of words or link words, *etc*… Which allow us to have a real control over what we're feeding the algorithm
- We can select what feature we are feeding the algorithm with : are we going to look only at the number of occurrences of words in Tweets, or at the sense of some sentences ? Are we going to include weights to the data, according to the number of followers of a Twitter account, the number of views or the amount of engagement (likes, comments…) of the Tweets ?

And all these different methods will influence the algorithm in different ways, which makes the project really interesting, but also really deep and complicated to implement, as there is also a sociology element added to the project.

We could even push the reflexion one step further, and consider that some Twitter accounts can be considered as "influencers", for example :

- A Tweet by Elon Musk, which literally crashed the DogeCoin (a cryptocurrency) value during the following hours.
- A company communicates on its will to accept Bitcoin payments.
- A country or influential politician declares its will to recognize Bitcoin (such as Salvador recently).

And others could be “followers” :

- A person who tries to analyse trends, without enough followers to be able to have a noticeable influence over Bitcoin’s value.
- A random Tweet “trolling”, or mocking Bitcoin without any repercussions.
- A person stating their gain, or warning others about their loss, again without a sufficient follower base.

However, this is a really advanced topic, and it could be the subject of a research paper by itself.

Obstacles for implementation

As said before, this operation, while in theory really interesting and promising, offers some unique challenges :

- The text in Tweets is usually informal, and written without any attention to the spelling of words or to the capitalization.
- The Twitter API requires programmers to meet some conditions in order to access Tweets older than a few days.
- The amount of Tweets you could request in a given time is limited, whatever your level of access to the API.
- The time constraint we had to develop the project was rather short for such a task.

Altogether, these elements put a stop to the development of that solution.

We also considered other options to increase the data dimensionality :

1. Use the value of altcoins (other cryptos than Bitcoin) as inputs, to maybe identify correlations between their evolution.
   However, it is hard to make the format of the dataset match, especially given the incredible lifespan of Bitcoin compared to other major altcoins.
2. Analyse news articles instead of Tweets, to have a more formal style of text to analyse, as well as reliable sources.
   But this does not eliminate all problems related to text analysis, and it is still really hard to gather the data from news websites : how to find them all ? How to treat the articles ? Plus, there is simply not often enough news articles related to Bitcoin, in comparison to Tweets that are way more instantaneous.

In the end, these options were not implemented due to the reasons enumerated before.

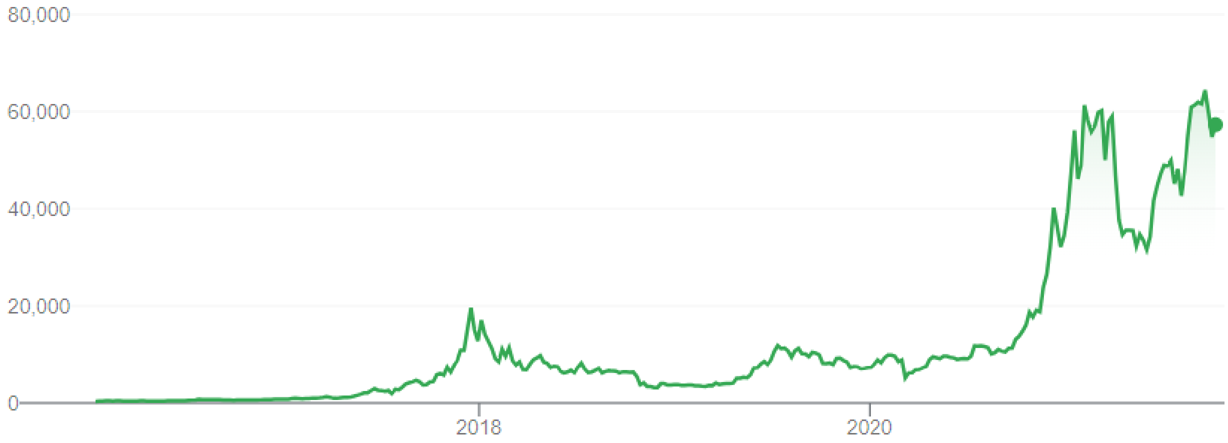
We also did not implement the other altcoins as features for the algorithm because of a lack of time. However, we did some research on the correlation between cryptocurrencies values, and we found some interesting results and tools. For example, the website https://www.blockchaincenter.net/cryptocurrency-correlation-study/ propose a complete correlation study between major cryptocurrencies and market indicators (Dow Jones, Gold…) over different time frames.

This led us to think that we could have slightly better results by implementing these in the program, without a dramatic increase in precision, because of the lack of insight given by these informations : if two cryptocurrencies are perfectly correlated, why would we want to analyze both if one is enough ?

Implemented methodology

You can check the code of the project in the following GitHub repository: https://github.com/javierherrer/cryptocurrency-predictor

The first step was importing the data, price of Bitcoin in Dollars, from the Yahoo Finance API in the period 2016-2021:



Then, the preprocessing step removes the NaN values and starts grouping the data, this is making groups of a specific *length* (hyperparameter) of dates. After that, the data is normalized using a MinMaxScaler. Then, the data is splitted in train and test sets (80% and 20%).

The RNN-LSTM is built with 3 hidden layers and the *hidden units* and *droput_rate* are set as hyperparameters. After that, the model is trained (*epochs* as a hyperparameter), predictions are made, and data is denormalized for plotting the graph.

All of the hyperparameters are searched using for loops and trying every combination comparing their results (loss function). This results are stored in a *scores-log.txt* file:

'**Length: 100 Epochs: 40 Units: 128 Dropout: 0.0': 0.0014809331623837352**,

'Length: 100 Epochs: 30 Units: 128 Dropout: 0.0': 0.0019011114491149783,

'Length: 80 Epochs: 40 Units: 128 Dropout: 0.0': 0.0028541090432554483,

'Length: 90 Epochs: 40 Units: 128 Dropout: 0.0': 0.0029099099338054657,

'Length: 90 Epochs: 30 Units: 32 Dropout: 0.0': 0.004055999219417572,

'Length: 80 Epochs: 30 Units: 128 Dropout: 0.0': 0.004541391506791115,

'Length: 90 Epochs: 30 Units: 128 Dropout: 0.0': 0.005040678661316633,

'Length: 100 Epochs: 40 Units: 64 Dropout: 0.0': 0.006123058497905731,

'Length: 80 Epochs: 20 Units: 128 Dropout: 0.0': 0.0066185640171170235,

'Length: 80 Epochs: 40 Units: 64 Dropout: 0.0': 0.008256030268967152,

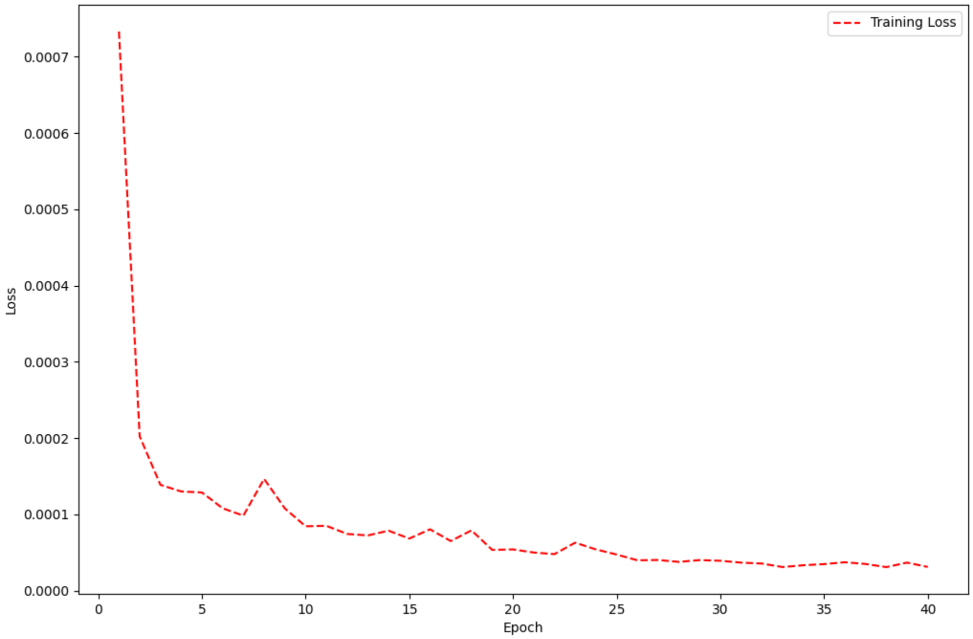
'Length: 80 Epochs: 30 Units: 32 Dropout: 0.0': 0.00843218993395567,

'Length: 80 Epochs: 40 Units: 32 Dropout: 0.0': 0.008438418619334698,

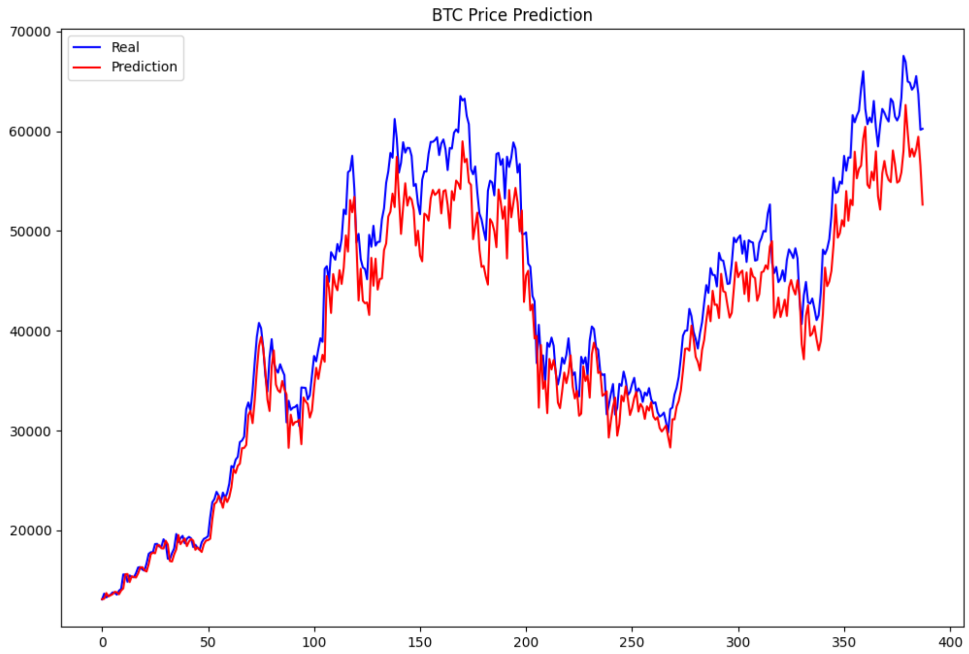
'Length: 100 Epochs: 20 Units: 128 Dropout: 0.0': 0.008761230856180191

Results without the twitter analysis

We run the model with the best hyperparameters found and obtaining the following training loss plot:



Review the imported data graph and remember we set the last 20% of it as a test set. Then check the results obtained comparing the real and predicted values:



Conclusion

Through the elaboration of this project an example of how to predict the cryptocurrencies behavior was shown, by the end of the analysis, we realized that new features could help solve the analysis presented in this paper, due to the fact that new features would increase the dimension of the data, other features that could be used are newspaper article analysis, companies official communication analysis, and other mathematical indicators (stock market indicators). Other improvements that could be made to this project would be a better hyperparameters search.