# Application of Deep Learning Towards Cryptocurrencies and Volatile Trading

**Flynn Traeger, Porter Weisberg**
Department of Computer Science
Stanford University
ftraeger@stanford.edu
porterjw@stanford.edu

## 1 Description of Research Problem

Broadly speaking, cryptocurrency trading is traditionally more volatile than standard trading. As a child born into the age of instant communication via the internet, cryptocurrencies are far more likely to fluctuate as a result of social media mentions and news headlines. We propose a two-fold attempt towards implementing an AI that better understands cryptocurrency trading. Firstly, with the availability of years of trading data, we plan to develop a deep learning tool that can best approximate gaps in the data we create as a proof of our model's efficacy for predicting data. Secondly, we are interested in attempting to include NLP sentiment analysis of social media mentions as an additional feature to better fine-tune our prediction model.

## 2 Related work

The study of economic and technological effects on the price of a cryptocurrency have been studied before, specifically, Li et al., [2017] came up with a hypothesis that cryptocurrencies respond to both traditional economic effects as well as technological effects. In an attempt to learn the process of developing a deep learning project we will pursue this same hypothesis as we attempt to meld elements of social mention data and economic data. Additionally, Allison Koenecke writes a tutorial [1] on how to apply Neural Networks to economic time series.

## 3 Datasets and Challenges

### 3.1 Early Findings

At our disposal we have a dataset containing financial data on 200 top cryptocurrencies, including daily open, high, low, total volume, and market cap values. This data extends daily from 2019 back several years. Some coins record up to six years of daily data other coins less. Additionally, nearly all major cryptocurrency exchanges output an API to access their own historical trading data. Further work can be done to determine which data is best for our use.

[1] https://koenecke.infosci.cornell.edu

One challenge we will face is the collection of social media mention and sentiment data. While historical data does exist, we have observed that it is usually locked behind a paywall. Down the road, we may run into issues with our existing data as some days do not contain every value. For instance, some days in 2019 for the cryptocurrency 'Ripple' is missing a market cap value.

### 3.2 Final Dataset

After conducting further research on different types of datasets and exploring what is freely available and accessible to us, we have settled on using data from data.world[2]. This document contains just over 13,500 data points formatted as shown in the screenshot below. The data is given for 10 different coins for the years 2014 to 2019 inclusive. We were planning on beginning by working with the data for Bitcoin and Ethereum for the early stages of our model.

| Currency | Date | Open | High | Low | Close | Volume | Market Cap |
|---|---|---|---|---|---|---|---|
| ripple | 24-Apr-19 | 0.321114 | 0.321282 | 0.296982 | 0.302318 | ########## | ########## |
| ripple | 23-Apr-19 | 0.323844 | 0.328396 | 0.320919 | 0.321222 | ########## | ########## |
| ripple | 22-Apr-19 | 0.322277 | 0.32935 | 0.320237 | 0.323934 | ########## | ########## |
| ripple | 21-Apr-19 | 0.328678 | 0.329627 | 0.318746 | 0.322449 | ########## | ########## |
| ripple | 20-Apr-19 | 0.331871 | 0.333213 | 0.324969 | 0.328476 | 931,570,799 | ########## |
| ripple | 19-Apr-19 | 0.337062 | 0.337147 | 0.329577 | 0.331902 | ########## | ########## |
| ripple | 18-Apr-19 | 0.335476 | 0.345289 | 0.335335 | 0.337065 | ########## | ########## |
| ripple | 17-Apr-19 | 0.327157 | 0.340327 | 0.322388 | 0.335453 | ########## | ########## |
| ripple | 16-Apr-19 | 0.320998 | 0.327308 | 0.319261 | 0.327218 | 799,796,781 | ########## |
| ripple | 15-Apr-19 | 0.328841 | 0.331551 | 0.31809 | 0.320913 | 934,107,164 | ########## |
| ripple | 14-Apr-19 | 0.32616 | 0.329421 | 0.324244 | 0.328809 | 746,765,729 | ########## |
| ripple | 13-Apr-19 | 0.326386 | 0.334444 | 0.323714 | 0.326031 | 976,688,780 | ########## |

Further to make this decision our time-series data, we decided to halt work towards finding further social media mention and sentiment data. This is explained further in the 'Midway Checkpoint' section below.

## 4 Proposed Methods

In class we briefly looked at types of neural networks and their strengths. Importantly, Recurrent Neural Networks are useful when working with data that is related to itself in an ordered series. Thus, when looking at a time-series of economic data, applying a RNN may make sense. There are more complicated versions of RNN's namely, a LSTM (Long short-term memory) that also excel at making sense of time series. Initially we are proposing the use of such an RNN, or a sequence-to-sequence model to predict a sequence of economic data from the dataset we have. Additionally, we would ideally like to include a feature that examines sentiment analysis of social media mentions over the time-series being examined by our sequence-to-sequence model.

In terms of what results we might expect, we see three potential applications and use cases. One, having trained a model we should be able to cut out a portion of data and have our model accurately predict what data went missing, this would likely best present on a graph or through a percent error metric. Secondly, once happy with a percent error metric we could additionally graph a future expectation of the cryptocurrency price movement. Finally, we could develop a classification model that takes both the predicted time-series and sentiment analysis of social media mentions of a particular cryptocurrency and classify whether it expects the price to go up or down.

## 5 Establishing Model Viability

One of the first decisions we made after conducting further research on datasets having cementing our initial project proposal was to halt work on finding further social media mention and sentiment data. This initial research had showed that most of this type of data that is available to be formatted for our needs is almost entirely pay-walled. We don't have any budget for purchasing this kind of data. There do exist APIs which we can utilise to try and collect this data for ourselves to then format and use. However, we decided it was more worth our time to shelve this as an extra piece of this project that we could return to if able, rather than delaying further work on the foundation of our model which focuses on using an LSTM Neural Network for cryptocurrency price predicitons. The

[2]https://data.world/pmohun/complete-historical-cryptocurrency-financial-data

social media mention data was always meant to be an additional layer so we don't feel as though the integrity of the project is compromised as a result of this decision, rather it is now more focused.

We then set about working on a model based on an LSTM Neural Network we came across during our research.[3] This is an older model used to similar effect as our intended model but on stock market data. The GitHub[4] gave us a starting point though it was heavily deprecated and required many changes to get it operational so that we could get any output using the sample data provided to us. This prompted us to search for alternative models and we were recommended N-BEATS by our supervising TA.

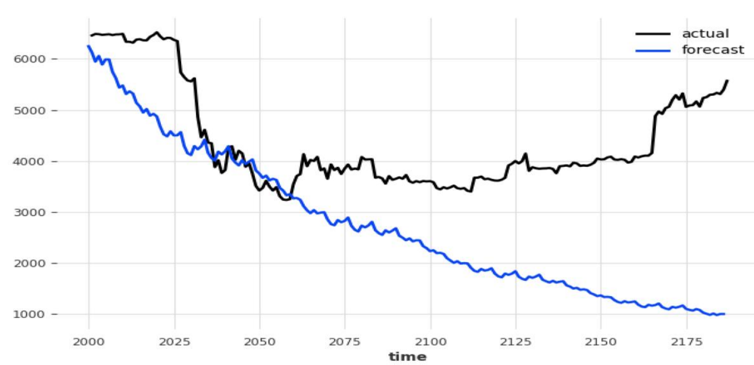
### 5.1 Moving to N-BEATS

While we were able to make changes to get the model to output correct predictions based on the provided sample data, after meeting with our assigned TA we started looking into N-BEATS[5] which seemed to be a more promising starting point for us. N-BEATS stands for Neural Basis Expansion Analysis for interpretable Time series forecasting. It is effectively a deep neural architecture based on backward and forward residual links and a very deep stack of fully-connected layers.[6]

The model takes a TimeSeries object as its input which we can create from a CSV to predict a future forecast. We formatted our data into a CSV and then used Darts[7] to convert it into a the TimeSeries primitive for the library to work with. The architecture is made up of stacks which each consist of K blocks. A block is a 4 layer FC Stack - made up of 4 FC layers and passing through a Relu function after each layer - which produces a backcast and a forecast. The forecast is used to contribute to that block's stack forecast and the backcast is given as stack residual as input to the next stack. Just one stack consists of many of these blocks following Double Residual Stacking principle.[8] These block forecasts give the stack forecasts which are then all used to give the global forecast - the model output. Loss is calculated using mean squared error on the global forecast and then used to update all gradients in the architecture.[9]

### 5.2 Initial Results

We have two early results to share. First, is a single prediction of the daily close amount of Bitcoin around the middle of 2019.




Second, is a series of predictions each predicting a week at a time over the same time period.

[3]https://www.altumintelligence.com/articles/a/Time-Series-Prediction-Using-LSTM-Deep-Neural-Networks

[4]https://github.com/jaungiers/LSTM-Neural-Network-for-Time-Series-Prediction

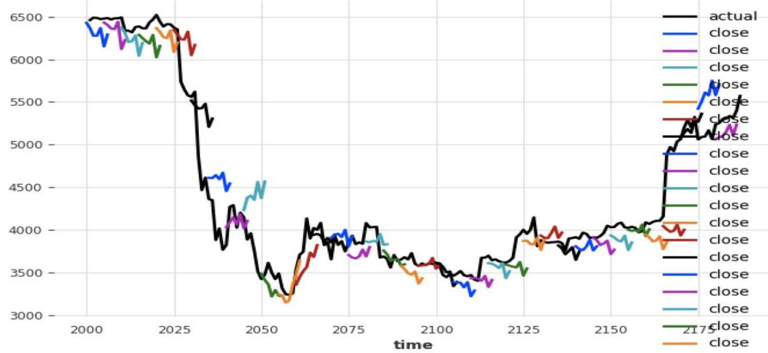
[5]https://unit8co.github.io/darts/examples/07-NBEATS-examples.html

[6]https://openreview.net/forum?id=r1ecqn4YwB

[7]https://unit8co.github.io/darts/quickstart/00-quickstart.htmlBuilding-and-manipulating-TimeSeries

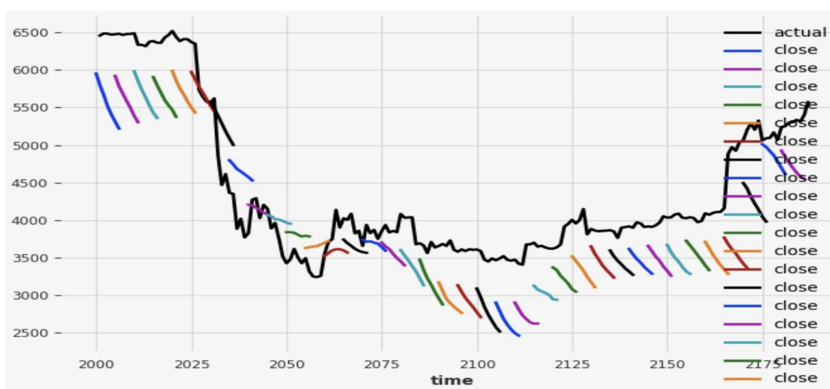
[8]https://arxiv.org/pdf/1905.10437.pdf

[9]https://tinyurl.com/2p97fmw7




# 6 Changes to Model and Training

One of the obvious takeaways from the results of testing shown above is that the model, history of 1 month and predicting 1 week, was biased and had learned a specific dip which it then kept repeating. Our next steps were to change the lengths of the lookback period and predict period for the training of our model. In endeavouring to find the correct balance between these two features we started tweaking them, settling on 1 day as the predict sequence length. We started with a lookback length of 7 days (which perhaps obviously produced worse, more biased results) before increasing it to 100 days, as shown below.




## 6.1 Multivariate Modeling

We continued varying the length of the lookback period many times and getting no better results than those shown above, we decided to move to a multivariate model. This was to incorporate the other data parameters we had available to us towards the training of the model, namely the values daily for bitcoin of: open price, close price, high price, low price, market cap, and volume.

This required more formatting of data, mostly in the form of splitting the data into separate CSVs that in our code we could construct separate TimeSeries objects to train the model on the different parameters. The way that N-BEATS handles and supports multivariate series is through "flattening the model inputs to a 1D series and reshaping the outputs to a tensor of appropriate dimensions"[10].

## 6.2 Future Forecasting

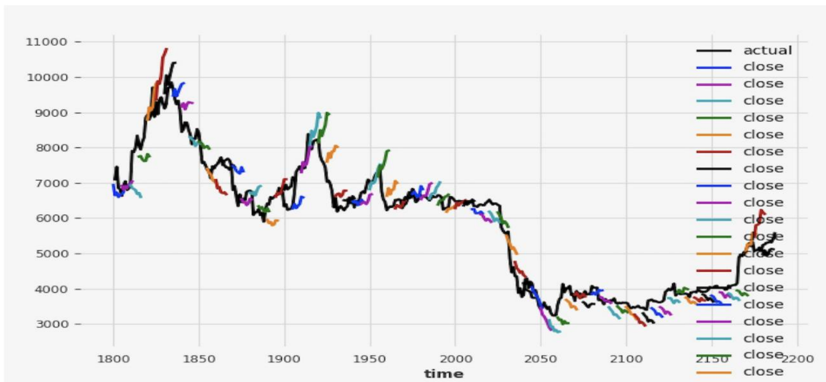
As part of our original goal of being able to produce a model with forecasting capabilities, we incorporated probabilistic forecasting into our multivariate model architecture. As with the aforementioned multivariate prediction, we had mixed success here and, while we were able to get reasonable enough outputs (when considered within the confines of our model and scope of the project), we do not have any way of properly checking the accuracy of this forecasting.

[10]https://unit8co.github.io/darts/generated_api/darts.models.forecasting.nbeats.html?highlight=multivariate

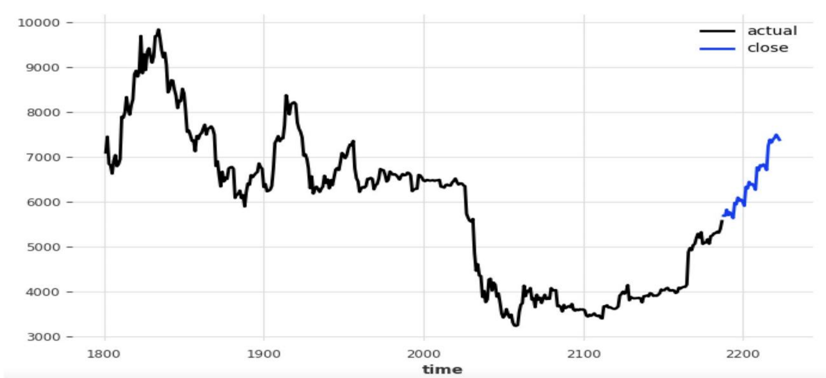
## 7 Results and Conclusion

As the title of our project suggests, the volatility of cryptocurrencies lends them to being very difficult to model. It was unfortunate that we had issues transfer learning between currencies as their own volatility did not seem to correlate to other currencies likely due to different causal factors which made our model worse. Also being unable to incorporate sentiment analysis data into this project was a shame, though from our experience in this we believe that it is likely this would have just added another layer of volatility into the mix and perhaps have hindered the model rather than helped it.

However, we were able to get results that we were happy with within the scope of what we'd set out to do and the confines and time of this particular project. Our final result graphs are below. The first being our historical prediction graph (like those we included previously).




The second result graph we have is for the forecast prediction from our model. The blue line below is our models prediction of the future price of Bitcoin based on its learning.




Ultimately, for our first foray into deep learning, this project was certainly challenging but we feel we came a long way from the beginning of it to a model that is producing results within a reasonable framework when compared to the historical data and producing a viable, if not entirely check-able, future forecast.

## 8 Contributions

This project was completed entirely by Flynn Traeger and Porter Weisberg with all portions being worked on together and in equal parts.