International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 5 May 2022, pp: 2159-2162 www.ijaem.net ISSN: 2395-5252

# Implementation of Bitcoin Price Predictor Using Artifical Intelligence

Aditya Agarwal, Anurag Panday, Adityanshu Singh

Submitted: 15-05-2022 Revised: 25-05-2022 Accepted: 28-05-2022

**ABSTRACT**
Bitcoin has recently become a menage name because of its unbelievable growth, unpredictable volatility, and attention-grabbing applications. This increase in interest within the public world has brought exaggerated interest within the information science world, best seen by the quantity of classifiers designed to see bitcoin worth fluctuations already. These classifiers, like Madan, Saluja and Zhao [1]'s as an example, have managed to achieve high accuracies within the past by specializing in strictly economic information relating to the Bitcoin network - average hashing, variety of transactions, etc. so as to tell apart this model, the main target was place virtually entirely on the result of stories on the value of bitcoin, beneath the belief that bitcoin's worth - particularly within the last year - is very influenced by "hype" that is oxyacetylene through what customers browse because the "state" of their world or of Bitcoin specifically. This model utilizes information processing techniques to investigate headline information for the past 2 years (2016 - 2017) to then feed a neural network to predict the fluctuation of bitcoin within the next day. With a restricted training/test set of ~640 examples, the model was able to attain sixty fourth check accuracy in predicting the sign of the amendment in bitcoin's worth within the next day.

## I. INTRODUCTION

As elaborate within the connected work section, there has been important effort place into predicting bitcoin costs exploitation historical economic information. Bitcoin has become Associate in Nursing more and more necessary a part of economic analysis as a result of it's become the figurehead for the complete cryptocurrency market. whereas this market of cryptocurrencies remains untested and unproved (as a viable currency) it's undeniably a strong construct backed by solid technology (blockchain). As a result, predicting this market is arguably as necessary as predicting the standard exchange. Besides this, the capability to form unbelievable returns on investment has conjointly been a key purpose of interest. The tested assumption of this model was the impact of "hype" on the value of bitcoin, particularly within the last year of very volatile behavior. This promotional material was measured exploitation sentiment analysis on a dataset of historical news headlines. employing a sentiment analysis library, TextBlob, i used to be able to generate a "sentiment" score for every headline, so weight this score by multiplying in Associate in Nursing "objectivity" score from constant library. This gave American state a feature indicative of "objective positivity" for every headline, that i used to be then able to average across all headlines for a given day to urge one input feature for each datum (a single day). This was my most important feature and consumed most of my development time (as delineated within the discussion section). as a result of I needed to even out my feature set and input a lot of to the network, I conjointly force in some terribly basic historical information on bitcoin, specifically: previous day's shut, previous day's volume, and therefore the label for the previous day (the closing price minus the gap value). I conjointly used the previous day's "objective positivity" score as a feature. In an effort to spice up my results, I conjointly adscititious information from Google trends, by propulsion analytics for the search term "Bitcoin," .

## II. RELATED WORK

Now that Bitcoin has cemented itself because the actual figure head for cryptocurrencies, the interest in predicting and modeling its behavior has skyrocketed. add this space has seen a dramatic increase in recent years, that i'll detail in brief. Associate in Nursing earlier model in 2015 by Alex Greaves and Benjamin Au [5] was solely able to bring home the bacon accuracy of fifty six exploitation neural networks to predict bitcoin worth fluctuation exploitation blockchain specific information. As mentioned earlier, Madan, Saluja and Zhao [1]'s model in 2014 showed rather more

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 5 May 2022, pp: 2159-2162 www.ijaem.net ISSN: 2395-5252

spectacular accuracy of ninety eight.7% in predicting bitcoin worth fluctuations on every day by day basis, and competent accuracy (50-55%) once narrowed to a ten minute widow. Hegazy and Mumford's paper [2] makes an attempt many various techniques for machine-driven bitcoin commercialism, finding that Boosted Trees provided the most effective check and coaching accuracy for his or her information set. a lot of recently, there has been some important work employing a Long Short Term Memory model to predict bitcoin worth fluctuations. JakobAungiers [3] was able to develop a model that closely matched the direction of Bitcoin's worth, however was even a lot of volatile than actuality fluctuations. Derek Sheehan's model [4], conjointly exploitation LSTM models, was able to bring home the bacon a larger illustration of actuality fluctuations, eventually inbound at a mean error of zero.04 and 0.05 for Bitcoin and Etherium, severally. whereas all of the models were able to bring home the bacon solid representations of the Bitcoin market, i believed the work done by Madan et al. [1]'s modelling was the foremost spectacular and thus the most effective place to focus my efforts. it might appear the "state of the art" (determined by the topics of recent papers) involves these LSTM models, however these (from my understanding) ar shaped specifically for statistic information rather more granular (by minute/second) than my information (daily).

## III. DATASET AND FEATURES

For sentiment score that my accuracy was concerning 4-7% lower for the check set, and also the average loss for This model utilizes 3 main sources knowledge|ofknowledge|of information} to detail historical Bitcoin data, historical news headlines, and search trends concerning Bitcoin. the primary 2 information sets (Bitcoinhistoricals and news headline historicals) were taken from Kaggle from the sets titled, "Cryptocurrency Historical costs," and "A Million News Headlines." though each informationsets provided data before 2016, I selected to limit the info set to 2016 and on, since I believed this was the amount most affected by "hype" and most characteristic of the intense inscrutable volatility we've got seen. each of those sets were on a usual, therefore my opening was to match the 2 datasets by change of integrity on the date. News headlines were separated into multiple rows of headlines for constant day, therefore when change of integrity I had to mix these headlines into one metric for every day. My initial efforts targeted on utilizing a "Bag of Words" illustration wherever every word is delineated by its own feature and also the information is that the frequency of every word within the summation of all headlines for that day. I conjointly tried to use a Word2vec model to get these options. However, as I delineate later within the Discussion section, this model was blemished partially as a result of it created a feature set so much larger (>40,000) than the scale of my example information set (~650). My final model utilizes a information science library, TextBlob [6], that provides ME with each a "sentiment" score Associate in Nursingd an "objectivity" score for every headline's text. I then averaged these scores over all the headlines for a given day, and utilised this as a feature. The Bitcoinhistoricals were abundant easier to figure with, as I merely used their values (normalized so as to be delineated efficiently) as options. In a shot to extend accuracy, I conjointly force in information from Google Trends by looking on the key word "Bitcoin" to do and model this "hype" behavior higher. the info set from this consisted of weekly "weights" for the recognition of this search term over the complete amount, that I then extended to every day and another as a feature. i used to be conjointly ready to use the Bitcoinhistoricals to get a label outlined to be one if the worth went up that day, and zero otherwise. By the top of it, Associate in Nursing example information seemed like this:

| Avg Sentiment Score | Prev. Day Bitcoin Close | Prev. Day Label | Prev. Day Sentiment | Prev. Day Adjusted Volume | Weighted Sentiment Score | Google Trends Rating | Label |
|---|---|---|---|---|---|---|---|
| 1.39543220e-02 | 4.10052000e+03 | 1 | -3.09499800e-03 | 3.76424000e+04 | 3.63496351e-01 | 11 | 0 |

FIGURE 1

**Methods** For this task, I utilised several of the canonical classification models out there to USA. To start, I used a supplying regression on my bag of words feature set to do and classify the sign amendment of bitcoin for the approaching day. supplying regression may be a supervised learning algorithmic rule that works by finding the simplest fitting weights to model a relationship between a collection of input options and their corresponding set of labeled outputs (of that there area unit 2 doable options: one or 0). This ideal weights area unit found by utilizing some type of gradient descent (stochastic or batch GD, for example) to reduce a loss perform or maximize a probability perform (of the info occurring). The canonical probability perform for supplying regression is that the log-likelihood, outlined as:

$$\begin{aligned}\ell(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))\end{aligned}$$

Where h(x) is outlined because the supplying perform of the weights dotted with the input. I finished up feeding this model a vector of >40,000 options, one for every doable word altogether of the headlines, that was then sculpturesque with >40,000 weights, one for every feature, which were then optimized exploitation random gradient descent to model the info I input. This model gave poor testing results, however nice coaching results, indicative of high bias and overfitting. Displeased with my results here, I switched to exploitation the sentiment analysis framework I antecedently mentioned to outline a feature, multiple linear relationships in one model, that was the hope in utilizing this model.which means my feature house is currently a way additional smart seven options massive. With this smaller feature house, I then ran a linear classifier to model this new feature set. Linear classification works equally to supplying regression except it produces a continual output, instead of a distinct output (like zero or one within the previous example). My hopes were to develop a classifier for the amendment in worth of bitcoin over every day, however this was met with terribly poor results, despite my amendment in feature set.

From this time i made a decision to turn over into exploitation neural networks to model this relationship. Specifically, I used the Deep Neural Network classifier provided by TensorFlow. This Deep Neural Network includes multiple layers of coupled "neurons" that every have weights from every input feature or previous layer to themselves. These weights area unit then dotted with the input feature values or the worth of the previous layer, then {passed through| skilled| older| morematured| moreexperienced|moreresponsible|more established|seasoned|knowledgeable|versed|capable |competent|skillful|well-versed|tried Associate in Nursingdtrue|gonethrough|had|undergone|saw|felt|r espondedto|suffered} an "activation function" to normalize the values. The hopes with this model is to seek out some relationship additional advanced than linear that may be delineated through this additional prolix model. The strictly larger model conjointly permits for way more fine standardization, which means we are able to categorical additional advanced relationships, or multiple linear relationships in one model, that was the hope in utilizing this model.

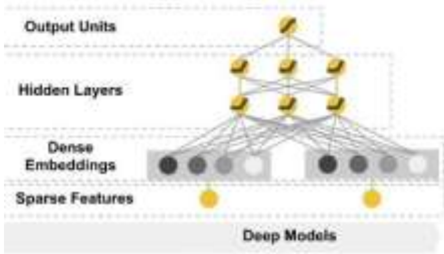


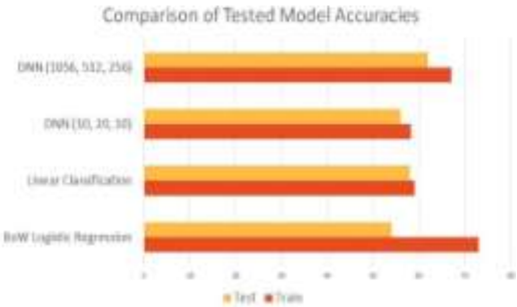
Figure 2.
Credit to: tensorflow.org/tutorials/wide_and_deep

## IV. EXPERIMENTS/RESULTS/MODEL

In an attempt to test my assumption that the fluctuations in bitcoin price could be found to be represented by the "hype" of news headlines, most of my work was spent in building features related to my news headlines dataset. The main metric I used to determine success was accuracy, since in this classification situation with a binary output, accuracy is very telling of the model's performance. A big rabbit hole I fell down was trying to model this by using the "Bag of Words" feature representation, where each word is represented by frequency used in each headline. I ran this with my logistic regression, but was only able to attain 54% test accuracy, and 73% training accuracy. This large difference in accuracies was likely due to the huge feature set overfitting to my small example set, which set off red flags. I tried a couple other methods to get this method to work, such as removing "stop words" like "the" from my dataset, and utilizing some frameworks like Word2vec to try and model these word features more efficiently. However, I was unable to improve these results in any significant way, and concluded that this representation was not really representative of the relationship I was trying to model.

So, despite investing significant time in this method, I decided to try a new route by utilizing a sentiment analysis framework, TextBlob, to provide the scores I mentioned earlier. With this new feature set, I ran linear classification which attained close to 0% accuracy since it was trying to model continuous values. I switched this to a logistic regression and attained 58% test accuracy and 59% training accuracy. While these accuracies were not good enough for my test (hence the rest of my expansion into using other models), I was happy to see that the bias was small and pointed to the model likely not overfitting.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 5 May 2022, pp: 2159-2162 www.ijaem.net ISSN: 2395-5252




From here, I moved to utilizing a Deep Neural Network with the same feature set. My initial implementation used a network model with 3 hidden layers, each having 10, 20, and 10 hidden units (neurons), respectively network model with 3 hidden layers, each having 10, 20, and 10 hidden units (neurons), respectively. With this preliminary and small network, I was only able to attain 56% test accuracy and 58.2% training accuracy. I then “beefed up” the network by adding more neurons in each hidden layer, up to 1024, 512, and 256, respectively, and was then able to attain only 0.69 average loss for the test set and 0.78 average loss on the test set, with 62% test accuracy, and 67% trainingaccuracy, the highest values I was able to achieve. The small difference between test and training accuracy also suggests minimal overfitting in this expanded network. I found that adding orremoving hidden layers from this only lowered my accuracy, and the same was true for the limited amount of changes in the counts of neurons per layer that I attempted. Most importantly, I found that when I removed the feature the test set without sentiment scores was 1.179, significantly more than with that feature, meaning the sentiment analysis feature had a significant impact. The accuracies of these models are summarized in the graph above.

## V. CONCLUSIONS

While my accuracy and loss values ne'er achieved the heights i used to be hoping for, I do believe i used to be able to with success take a look at my assumption that the worth of Bitcoin is influenced by the publicity of reports headlines. i used to be able to bring home the bacon the bottom loss on the take a look at set and also the highest accuracy after I used a Deep Neural network, that was to be expected compared to less complicated supplying and linear regressions since this deep network is delineate as layering multiple of those less complicated algorithms along and interleaving their results. If given longer and resources, i'd prefer to investigate different cubic centimetre techniques, like the LSTM models I mentioned within the connected Works section, or increasing my feature set to higher approximate bitcoin specific sentiment exploitation custom informatics techniques. I do believe that the publicity of the fashionable news very influences the worth of bitcoin, however I can’t deny the very fact that the fundamental economic options associated with block level transactions give a awfully representative, if less representative, model of the behavior of bitcoin’s worth. Thus, i believe a model combining all of those options employing a additional specific formula would be able to give a good illustration of this market, if given the correct knowledge and time to develop.

## REFERENCE

[1]. Madan, S. Saluja, and A. Zhao, “Automated Bitcoin Trading via Machine Learning Algorithms.” [Online]. Available:http://ai2-s2 pdfs.s3.amazonaws.com/e065/3631b4a476abf5276a264f6bbff40b132061.pdf. [Accessed: 14-Dec-2017].

[2]. Hegazy, K. and Mumford, S. (2016), “Comparative Automated Bitcoin Trading Strategies.”

[3]. J. Aungiers, “Multidimensional LSTM Networks to Predict Bitcoin Price,” JakobAungiers, 15-Jul-2017.

[4]. D. Sheehan, “Predicting Cryptocurrency Prices With Deep Learning,” dashee87.github.io, 19-Nov-2017

[5]. A. Greaves and B. Au, “Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin,” stanford.edu, 08-Dec-2015