Original Article
ISSN (Online): 2454-1907

International Journal of Engineering Technologies and Management Research
May 2021 8(5), 20–28

BITCOIN PRICE PREDICTION USING MACHINE LEARNING

Mrs Vaidehi M[1], Alivia Pandit[2], Bhaskar Jindal[2], Minu Kumari[2] and Rupali Singh[2]

[1] Assistant Professor, Department. of Information Science and Engineering, DSCE, Bangalore, India
[2] Department of Information Science and Engineering, DSCE, Bangalore, India

ABSTRACT

In this paper, we use the LSTM version of Recurrent Neural Networks, pricing for Bitcoin. To develop a better understanding of its price influence and a common view of this good invention, we first give a brief overview of Bitcoin again economics. After that, we define the database, including data from stock market indices, sentiment, blockchain and Coinmarketcap. Further in this investigation, we demonstrate the use of LSTM structures with the series of time mentioned above. In conclusion, we draw the Bitcoin pricing forecast results 30 and 60 days in advance.

Keywords: Bitcoin, Crypto Currency, Machine Learning, Blockchain, Long Short Term Memory(LSTM), Recurrent Neural Network(RNN), Prediction

1. INTRODUCTION

With the appearance of Bitcoin 10 years ago the globe economist, albeit in small numbers, is flexible and responsive. Bitcoin introduced itself as a program that solved the Double Spend problem Nakamoto and Shah (2017) , a preferred issue with Digital Cash systems. However, the impact in the coming years was great. Distributed Ledger Technologies (DLT), Intelligent Agreements, Cryptocurrencies, etc. it's all supported by the thought of "Bitcoin". This was identified, during a separate power division mixed with intuitive motive. On the opposite side of the spectrum, and data is taken into account nowadays, over time with a major increase in hardware efficiency, Machine learning continues to be used. As a result, we tend to predict the worth of Bitcoin, while the dynamic isn't not only on Bitcoin exchanges but also on finance markets generally.

Received 5 May 2021
Accepted 20 May 2021
Published 31 May 2021

Corresponding Author
Mrs Vaidehi M, vaidehi-ise@dayanandasagar.edu

DOI 10.29121/ijetmr.v8.i5.2021.953

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

How to cite this article (APA): M. V. M., A. P., B. J., M. K., & R. S. (2021). Bitcoin price prediction using machine learning. *International Journal of Engineering Technologies and Management Research, 8*(5), 20-28. doi: 10.7821/ijetmr.v8.i5.2021.953

2. LITERATURE SURVEY

We have all considered where bitcoin costs will be one year, two years, five years or even 10 years from now. It's really difficult to anticipate however each and every one of us loves to do it. Tremendous measures of benefits can be made by purchasing and selling bitcoins, whenever done accurately.. It has been proven to be a fortune for many people in the past and is still making them a lot of money today. But this doesn't come without its downside too. If not thought of and calculated properly, you can lose a lot of money too. You should have an incredible comprehension of how and precisely why bitcoin costs change (organic market, guidelines, news, and so forth), which implies you should realize how individuals make their bitcoin predictions. Considering these things (supply and demand, regulations, news, etc.), one must also think about the technology of bitcoin and its progress. This aside, we now have to deal with the technical parts using various algorithms and technologies which can predict precise bitcoin prices. Although we came across various models which are currently present like Biological neural networks. (BNN), Recurrent neural network (RNN), Long short-term memory (LSTM), Auto regressive integrated moving average (ARIMA), etc. with machine learning and deep neural network concepts. Normally a time series is a sequence of numbers along time. This is due to the fact that this being a time series data set, the overall data sets should be split into two parts: inputs and outputs.Moreover, LSTM is great in comparison with the classic statistics linear models, since it can very easily handle multiple input forecasting problems.

In the approach which we are following, the LSTM will use the previous data to predict bitcoin prices 30 days ahead of it's closing price. In the approach used by us, we implement Bayesian optimized Recurrent Neural Network (RNN) and a Long Short Term Memory (LSTM) network. The highest classification accuracy is achieved by LSTM with the accuracy of 52% and a RMSE of 8%. Presently we execute the famous Auto backward incorporated moving normal (ARIMA) model for time arrangement gauging as a correlation with the profound learning models. The ARIMA forecast is out performed by the nonlinear deep learning methods which performed much better. Finally both the profound learning models are benchmarked on both a GPU and CPU. The training time on the CPU is outflanked by the GPU execution by 67.7%. In the base papers selected by us, the author collected a data set of over 25 features relating to the bitcoin price and payment network over a period of five years, recorded on a daily basis were able to predict the sign of the daily bitcoin price change with an incredible accuracy of 98.7%.

In the second period of our examination we are just focusing in on the bitcoin price information alone and utilized information at 10 minutes and 10 seconds time frame.. This is due to the fact that we saw an incredible opportunity to precisely evaluate price predictions at various levels of granularity and noisiness are modelling. This resulted in incredible results which had 50 to 55% accuracy in precisely predicting the future bitcoin price changes using 10 minute time intervals.

3. DATA PREPROCESSING

3.1 DATA GATHERING

Daily data for the four channels has been monitored since 2013. First, the Bitcoin price history, from which it is extracted The coin market is the head of the market with its open API. Second, data from Blockchain included, especially we prefer standard block size, user address number, the amount of production, and the number of miners. We find it objectionable to have Blockchain data, given the endless measurement problem, on the other hand, the number of accounts, by definition related in price movements, as the number of accounts increases, it could mean more transactions that take place (perhaps by exchanging different parties and not just by transferring Bitcoins to another address), or by signaling more users joining the network. Third, in the emotional details, we find that over time the term 'Bitcoin' was used by the PyTrends library. Finally, two indicators are considered, those of S&P 500 and Dow and Jones. Both are refundable Yahoo Finance API. In total, this makes 12 features. Pearson interaction between symptoms is shown in Figure 2. Some attributes do not exist closely related, for example, financial indicators suitable for each other, but not for any of the attributes associated with bitcoin. Also, we see that Google Trends are related to Bitcoin transactions.

3.2 DATA CLEANSING

From exchange data, we look only at the related Volume, Close, Unlock, higher prices, and market capitalization. In all data sets, if the NaN values are found to be correct there, it is replaced by a description of the appropriate attribute. After this, all data sets are merged into one, according to the magnitude of the time. If we look at the Bitcoin price movement during the period from 2013 to 2014, we have seen fit to remove data points prior to 2014, which is why the details that will be transferred to the neural network are dormant from 2014 to September 2018.

3.3 DATA NORMALIZATION

Deciding how to get used to the timeline, especially finance is by no means easy. What else, as a sixth rule, the neural network must load data taking large amounts of different data (referring to different time series scales, such as exchange rate, and Google Trends). Doing so can create major gradient updates that will prevent the network from changing. Doing reading easy on the network, data should have the following features:

- Take small values- Typically most values should be in range 0-1.
- Be homogeneous- That is, all features should take values at roughly the same range.

Min-MaxScaling, where the data inputs are mapped on a number from 0 to 1:x'=x-min(X)/max(X)-min(X)

3.4 DATA TRAINING AND SPLITTING

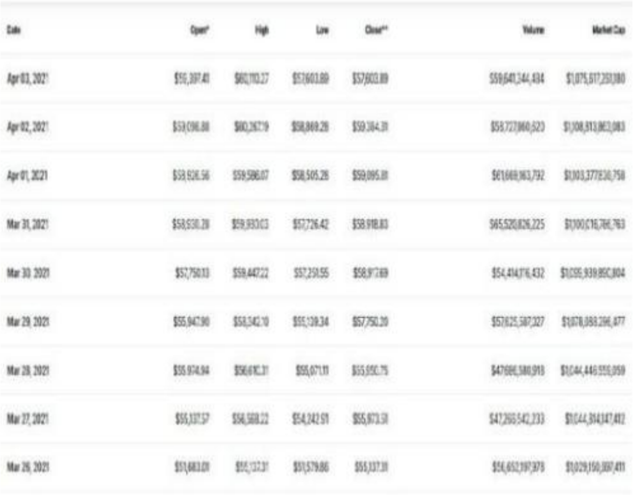
We first wanted to predict next year, but this may mean, that data from 1 Jan 2018 until September 2018 will be used for testing, the downside of this, is actually a major slope in 2017, which could make the neural network learn this pattern as the final input, and the prediction of the year 2018 would not have been so sensible. So off we go training data from 2014-01-01 to 2018-07-05, this leaves us with 2 months to predict, while predicting two months, the data set is split premature leave room 2 months: 2018-06-01. Each training set and test set is built on reinstallation output features.

4. MACHINE LEARNING PIPELINE

4.1 SOFTWARE USED

In the Deep Learning backend program, we select Tensorflow and Keras as the front end layer for building neural networks faster. Pandas is mainly used for data-related activities, Numpy is used for matrix/vector performance and for keeping data and training sets, Scikit-learn (also known as sklearn) is used to make min-max standardization. Finally, Plotly is used to display charts.

4.2 TIME SERIES DATA



Date	Open*	High	Low	Close**	Volume	Market Cap
Apr 03, 2021	$59,397.41	$60,110.27	$57,603.89	$57,603.89	$59,641,344,484	$1,075,617,253,180
Apr 02, 2021	$59,098.88	$60,267.19	$58,869.28	$59,384.31	$58,727,860,620	$1,108,313,863,083
Apr 01, 2021	$58,926.56	$59,586.07	$58,505.28	$59,095.81	$61,669,163,792	$1,103,377,830,758
Mar 31, 2021	$58,930.28	$59,930.03	$57,726.42	$58,918.83	$65,520,826,225	$1,100,016,786,763
Mar 30, 2021	$57,750.13	$59,447.22	$57,251.55	$58,917.69	$54,414,116,432	$1,099,939,850,804
Mar 29, 2021	$55,947.90	$58,342.10	$55,139.34	$57,750.20	$57,625,587,327	$1,078,083,296,477
Mar 28, 2021	$55,974.94	$56,610.31	$55,071.11	$55,950.75	$47,686,580,918	$1,044,446,555,059
Mar 27, 2021	$55,137.57	$56,568.22	$54,242.91	$55,973.51	$47,266,542,233	$1,044,814,347,412
Mar 26, 2021	$51,683.01	$55,137.31	$51,579.86	$55,137.31	$56,652,197,978	$1,029,150,887,411

Figure 1

Usually a series of time sequences of numbers in order time. LSTM sequence prediction works as a controlled algorithm, unlike its auto-encoder version. As such, the complete database should be separated from the input and results. Moreover, LSTM is good in comparison with older models of classic statistics, because it can be easier to manage multiple installation prediction problems. On our way, LSTM will use

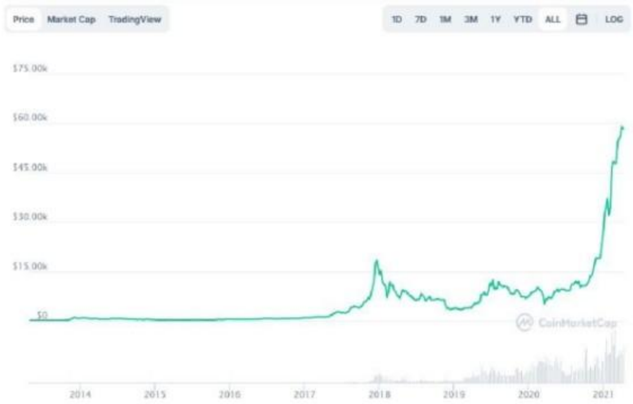


Figure 2

previous data to predict 30 days before closing price. First, we must decide which prediction that we will be able to access in the last few days. This number we call window size. We have selected 35 days in the event of a monthly forecast, too 65 days in that two-month forecast, so the input data set will be a tensor containing matrices size 35x12 / 65x12 respectively, for us it has 12 features, and 35 lines in each window. Therefore The first window will have 0 to 34 rows (python is zero-indexed), the second from 1 to 35, and so on. Another reason to choose window length is to have small window leaves patterns that may appear in a long sequence. Output data looks not only at the size of the window but also the width of the forecast which for us is 30 days. The output database starts from line 35 to the end and is made up of connectors of length 30. The width of the forecast also determines the output size.

4.3 LSTM IMPLEMENTATION

A key feature of feed networks is that they do not save memory. Therefore each input is processed independently, without the saved state in the middle of the input. Given that we are dealing with a series of times where information from the previous Bitcoin price is required, we should keep track of future events. The building that provides this is the Recurrent neural network (RNN) associated with output has an automatic loop. So the window we provide as input is processed sequentially rather than one step. However, when the measure of time (size of the window) The largest (most common) gradient is the smallest / largest, leading to a condition known as the disappearance/explosion of the gradient respectively. This problem occurs over time backpropagate optimizer and will activate the algorithm, while the tools are unlikely to change at all. RNN variation reduces the problem, i.e. LSTM and GRU. The LSTM layer adds other data-carrying cells to multiple timesteps. Cell status is a horizontal line from Ct-1 to Ct, and its value lies in capturing long-term or short-term memory. The LSTM effect is adjusted by the government to taste the cells. And this is important when it comes to predicting based on historical context, and not just lastly. LSTM networks can remember input using a loop. These logs are not in RNN. On the other hand, as time goes on, it is less likely that the next result will depend on installation being very old, so forgetting is necessary. LSTM achieves this by learning when to remember and when you forget, by their gates you forget. We will mention soon not to view LSTM as a black-box model.

- Forget gate: ft = σ(WfStt 1 + WfSt)
- Input gate: it = σ(WiStt 1 + WiSt)
- Output gate: ot = σ(WoStt 1 + WoSt)

4.4 ARCHITECTURE OF LSTM

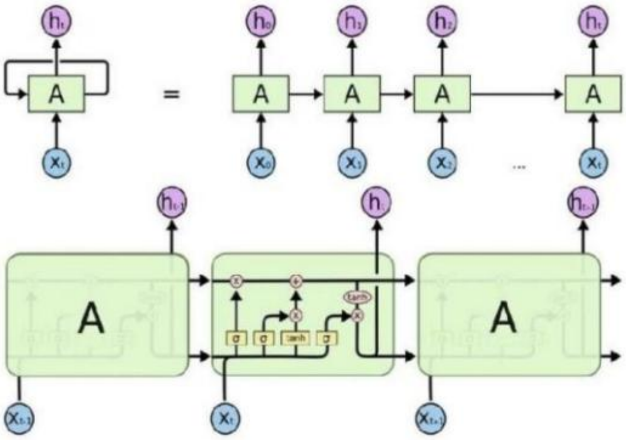


Figure 3

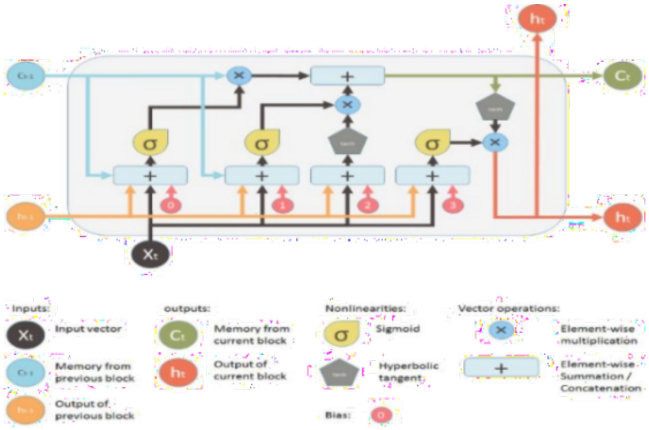
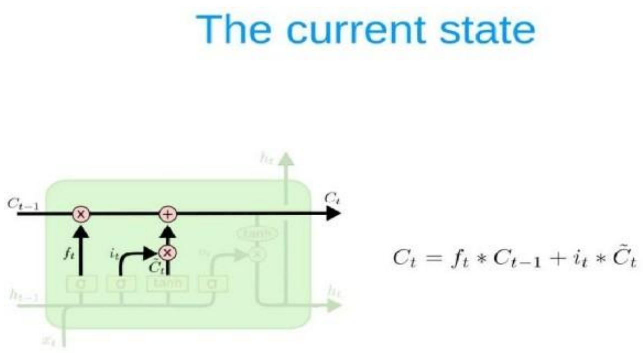


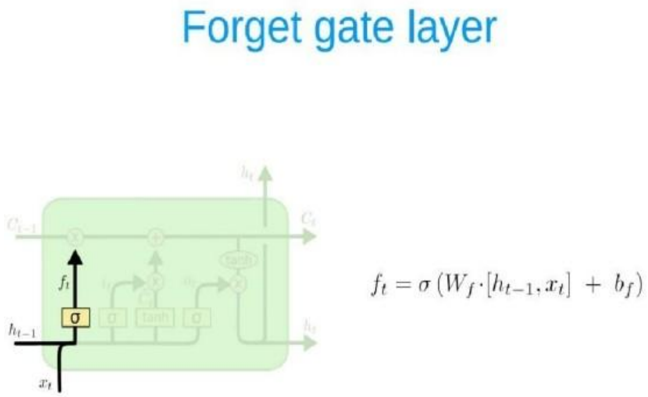
Figure 4



The current state

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

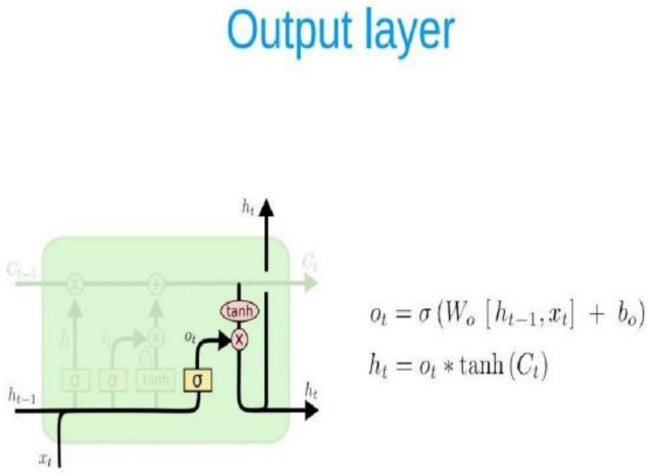
Figure 5



Forget gate layer

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure 6

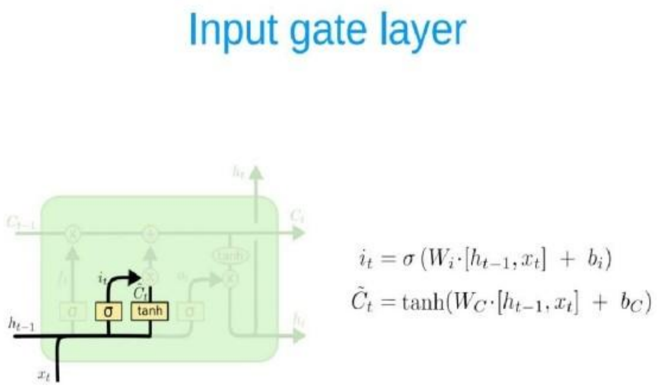


Output layer

$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 7



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Figure 8

4.5 ARCHITECTURE OF LSTM

We have used the Sequential API for Keras, rather than which works. The complete construction is as follows:

- LSTM Layer: LSTM Layer is inside one, and all the gates, mentioned earlier have already been used by the Keras, with auto-sigmoid automation [Keras2015].

LSTM parameters are the number of neurons and the input mode as described above.

- Dropout Layout: This is usually used before a thick layer. As for Keras, dropout can be the case added behind any hidden layer, for us, in the background LSTM.
- Dense Layer: This is a standard layer that is fully integrated.
- Background Layout: Because we solve a retreat problem, the last layer should provide a combination of the line performance for the previous layer and weight vectors. Either way, it is possible to transfer as a parameter to the previous dense layer.

5. RESULTS AND ANALYSIS

In this section we show the results of our LSTM model. It was noted during training that the higher the batch size the worst the prediction on the test set. Of course this is no wonder, since the more training, the more prone to overfitting the model becomes . While it is difficult to predict the price of Bitcoin, we see that features are critical to the algorithm, future work includes trying out the Gated Recurrent Unit version of RNN, as well as tuning, on existing hyper-parameters. Below we show the loss from the Mean Absolute Error function, when using the model to predict the training and test data.

6. CONCLUSION

All in all, predicting a price-related variable is difficult given the multitude of forces impacting the market. Add to that, the fact that prices are by a large extent dependent on future prospects rather than historic data. However, using deep neural networks has provided us with a better understanding of Bitcoin, and LSTM architecture. The work in progress, includes implementing hyperparameter tuning, in order to get a more accurate network architecture. Also, other features can be considered (although from our experiments with Bitcoin, more features have not always led to better results). Microeconomic factors might be included in the model for a better predictive result. Anyway, maybe 6 Conclusions All in all, predicting a price-related variable is difficult given the multitude of forces impacting the market. Add to that, the fact that prices are to a large extent depended on future prospects rather than historic data. However, using deep neural networks has provided us with a better understanding of Bitcoin, and LSTM architecture. The work in progress, includes implementing hyperparameter tuning, in order to get a more accurate network architecture. Also, other features can be considered (although from our experiments with Bitcoin, more features have not always led to better results). Microeconomic factors might be included in the model for a better predictive result. Anyway, maybe the data we gathered for Bitcoin, even though it has been collected through the years, might have become interesting, producing historic interpretations only in the last couple of years. Furthermore, a breakthrough evolution in peer-to-peer transactions is ongoing and transforming the landscape of payment services. While it seems all doubts have not been settled, time might be perfect to act. We think it's difficult to give a mature thought on Bitcoin for the future.

REFERENCES

&amp; Santhosh, G, S., & ', . (n.d.). A Relative Study on Bitcoin Mining. *Relative Study on Bitcoin Mining", Imperial Journal of Interdisciplinary Research*.

Garcia, D., & Schweitzer, F. (2015). Social signals and algorithmic trading of Bitcoin. *Royal Society Open Science*, *2*(9), 150288–150288. Retrieved from https://dx.doi.org/10.1098/rsos.150288 10.1098/rsos.150288

Jang, H., & Lee, J. (n.d.). An Empirical Study on Modelling and Prediction of Bitcoin Prices with Bayesian Neural Networks based on Blockchain Information. *An Empirical Study on Modelling and Prediction of Bitcoin Prices with Bayesian Neural Networks based on Blockchain Information*.

Kathyayini, R. S., Jyothi, D. G., & Crypto, . . (n.d.). Currency Price Prediction using Machine Learning". *International Journal of Advanced Research in Computer and Communication Engineering*.

Nakamoto, S., & Shah, . D. (2017). Trading Bitcoin and Online Time Series Prediction. *NIPS 2016 Time Series Workshop*.

Struga, K., & Qirici, O. (n.d.). Bitcoin Price Prediction with Neural Networks. *Bitcoin Price Prediction with Neural Networks*. Retrieved from http://ceur-ws.org/Vol-2280/paper-06.pdf