International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 1, ISSN No. 2455-2143, Pages 502-506
Published Online May 2020 in IJEAST (http://www.ijeast.com)

PRICE PREDICTION OF BITCOIN USING MACHINE LEARNING

Reshma Sundari Gadey, Nikita Thakur, Naveen Charan
UG Student, Department of CSE, Institute of Aeronautical Engineering, Dundigal,
Telangana, INDIA.

R. Obulakonda Reddy
Associate Professor, Department of CSE, Institute of Aeronautical Engineering,
Dundigal, Telangana, INDIA.

Abstract - With the development Machine learning and AI- assisted trading has gained interest in the past few years. To bring out the abnormal profits from the cryptocurrency market, we use this machine learning and AI-assisted trading. We store the daily data for a certain period. With the strategies assisted by state-of-the-art algorithms we obtain great outcomes. With the help of simple algorithms and architecture, the outcomes made the growth in the cryptocurrency market. The cryptocurrency has become popular in 2017 because of the growth in market capitalization. More than 1500 crypto currencies are actively trading in today's scenario. The crypto currency can be generated and be used for online transactions. Bitcoin is a cryptocurrency technology. The value of Bitcoin keeps varying every second. Therefore, to predict the value of bitcoin price here, we use the LSTM Architecture. With the help of this architecture, we are trying to prove this LSTM architecture provides more accurate results than any other machine learning algorithms and architecture.

Keywords: Machine learning, Bitcoin, Price prediction, LSTM.

I. INTRODUCTION

In 2017 due to repeated exponential growth in capital, the popularity for crypto currencies had taken uptrend. In the present scenario, there are more than 1,500 actively traded crypto currencies which capitalize over $300 billion, in Jan 2018 the market capital is totaling more than $800 billion. In between 2.9 and 5.8 millions of private investors and institutions in different networks, having access to market had become simple. Most of the crypto currencies are bought of online exchanges, also the volume of daily online exchanges is up to $15 billion. Due to the emerging of 170 hedge funds in crypto currencies, the bitcoin future is having demand for trading and hedging Bitcoin. There are several conspiracies regarding the precise reasons behind this volatility and these theories are also used to support the prediction reasoning of crypto prices, particularly of BTC. The parameters are important to predict the future of crypto currencies, our way of prediction approaches this issue from a different perspective, particularly, that of algorithmic trading. We simply plan to use numerical historical data to train a recurrent neural network (RNN) to predict BTC prices. A recurrent neural network is a type of artificial neural network in which the connections are made between the nodes form the directed graphs along a temporal sequence. LSTM is an artificial RNN architecture which is used in deep learning which not only processes the single data points but the total data.

II. LITERATURE SURVEY

(2016) Can we predict the winner in a market with network effects? Competition in cryptocurrency market [3]:

The authors analyzed how networks affect the cryptocurrency market. This was done by taking the exchange rates of different crypto currencies. In this, it was proved that the bitcoin was the most dominant in the market. The data that was taken into consideration was consistent with strong network effects.

(2015) Automated Bitcoin Trading via Machine Learning Algorithms[4]:

The authors tried to use a machine learning algorithm i.e. custom algorithm which includes both the random forest and generalized linear model for the prediction of the bitcoin price. The dataset is taken consists of 25 features over a period of five years. It was proved that they had gained an accuracy of 50-55% in predicting the future price. The data was considered with a time interval of 10 minutes.

(2013) API design for machine learning software: experiences from the scikitlearn project [5]:

The authors describe the famous and the most used machine learning library. This tool is very useful in implementing machine learning algorithms. The authors tell us how simple and efficient the library is and describes the advantages of the sci-kit learn. It also deals with the implementation of the library to the python ecosystem and describes the hardships faced by the developers while using this tool. The other libraries that are needed in the implementation of the machine learning algorithm are Keras and TensorFlow.

(2015) Keras: Deep learning library for theano and tensorflow [6]:

The authors have very well explained the usage of the tools. The Keras library is implemented on the python ecosystem and supports the running of the TensorFlow in the backend. Keras is the neural network library and was developed to enable fast experimentation. This library supports both convolution networks and recurrent networks.

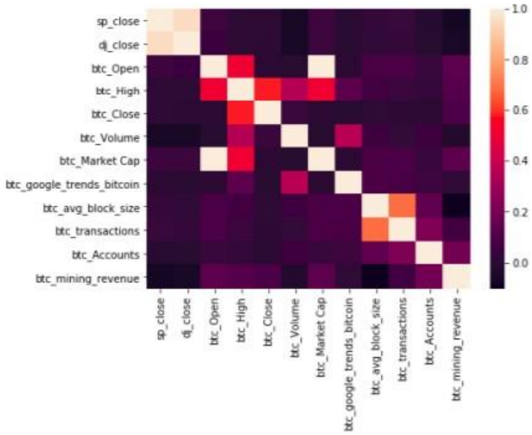
(2016) Tensorflow: Large-scale machine learning on heterogeneous distributed systems [7]:

The author showed the comparison in predicting the value of bitcoin by using various regression models with the sci-kit learn and Keras libraries. By the end of the comparison, the best results obtained are that the R-square was high and the mean squared error was low.

III. IMPLEMENTATION METHODOLOGY

Data pre-processing:

Data collecting: - To start with, the Bitcoin value history, which is separated from the Coin market cap through its open API. Furthermore, information from Blockchain is accumulated. We thought that it was irrational to have some Block chain information, given the unending scaling issue, on the other hand, the quantity of records, by definition, is connected to the cost developments, since an expansion in the number of records either implies more exchanges occurring or it is an indication of more clients joining the system. Thirdly, for the conclusion information, we acquire the word 'Bitcoin' utilizing PyTrends library.



Data cleaning: - From the collected data we consider only the necessary columns like Volume, Close, Open, High costs, Market capitalization for the future process. if NaN values are found then they are replaced with the mean of the particular attribute. After this, all datasets are combined into one. The data that is passed is from 2014 to 2018.

Data normalization: - Selecting the approach for normalizing a time series is by no means a smooth process. The data which is considered is very huge and is a heterogeneous data. In this process, we come across large gradient updates which will prevent the converging of the network. Changing the data into the below mentioned format makes learning less difficult.

Take small values - Most values have to be within the range of 0-1.

Homogeneous - That is, all parameters should have a similar type and range.

Normalization Methods:

Min-Max Scaling, where the information inputs are mapped from 0 to 1:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}$$

Mean Normalization, which makes data to have a range between -1 and 1 with a mean of 0:

$$x' = \frac{x - \mathrm{mean}(X)}{\max(X) - \min(X)}$$

Z-Score (Standardization), where the parameters are replaced with their mean of 0 and a standard deviation of 1:

For our project, we use Min-Max Scaling and change the features on a scale from 0 to 1. Min-Max Scaling works perfectly for this large datasets.

Machine Learning Pipeline

In this segment, we make time-series data adaptable for supervised machine learning problems. The price prediction is handled as regression instead of classification, and we show how LSTM can be utilized in such cases.

Softwares used: - In this, we use Tensorflow, Keras, Pandas and Numpy. For Deep Learning backend framework we use Tensorflow and Keras as the front-end for neural networks. Pandas for data associated tasks, Numpy for matrix or vector operations and for storing training and test sets, Scikit-learn for the min-max normalization. Plotly for the representation of the data in the form of charts.

Time series data: - Generally a time series is a series of numbers along time. LSTM for series prediction acts as a supervised set of rules in contrast to its auto encoder rendition. The dataset should split into inputs and outputs. In addition, LSTM is the best in comparison with classic statistics linear models, since it can easily handle different data. In our project, the LSTM will utilize the past data to predict future data. To begin with, we should decide on what number of previous days one forecast will have access to. This range is called window size. The matrix formed will be depending on the values that are considered for the prediction. Another purpose behind picking this window length is that a small window leaves out patterns which may appear in a longer sequence. The output data takes window length and the prediction range. The prediction range additionally determines the output size for the LSTM network.

Split into training and test data: - This step is one of the most significant, particularly in the instance of Bitcoin. We want to predict values, the data from June 2017 would be utilized for testing. We consider 31942 training set values and 15734 test set values. Each training set and test set contains input and output features.

Turn data into tensors: - LSTM expects that the input which is given should be in the shape of the 3- dimensional vector of float values. A key component of tensors is their shape, which in Python is a tuple of integers representing the dimensions of it along the 3 axes. In LSTM the input layer is determined from the information shape, this includes the samples, window size and the number of features.

$$x' = \frac{x - \mathrm{mean}(x)}{\rho}$$

Implementation of LSTM: - The main component of feed forward Networks is that they try not to hold any memory. So each input is independent. Given that we are having a time series where data from past Bitcoin cost are required, we ought to keep up some data to foresee what's to come. An architecture presenting this is the recurrent neural system (RNN) which along with the output has a self-directing loop. So the window we give as the input gets prepared in a sequence. When the time step (size of the window) is huge (which is frequently the situation) the gradient becomes small or large, which leads to vanishing or exploding gradient respectively. This happens while the optimizer back propagates, and will make the algorithm run. RNN varieties relieve the issue, in particular, LSTM and GRU. The LSTM layer includes cells which carry data across numerous time steps. The cell state is the horizontal line from C_{t-1} to C_t, and its significance lies in holding the long term memory or short term memory. The yield of LSTM is regulated by the state of these cells. Furthermore, this is significant when it comes to predicting historic context, instead of having only the last input. LSTM networks use loops to remember the inputs. These loops are not present in RNN. As time passes it becomes less likely that the output depends on the very old inputs hence the values can be deleted. LSTM achieves this by using the forget gates.

Forget gate: $f_t = \sigma\ (W_f S_{t-1} + W_f S_t)$

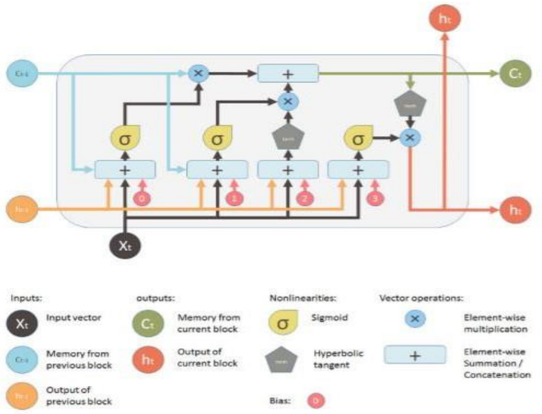
Input gate: $i_t = \sigma\ (W_i S_{t-1} + W_i S_t)$

Output gate: $o_t = \sigma\ (W_o S_{t-1} + W_o S_t)$

Intermediate Cell State: $C^{\sim} = \tanh\ (W_c S_{t-1} + W_c X_t)$

Cell state (memory for next info): $c_t = (i_t * C^{\sim}\ t) + (f_t * c_{t-1})$

Calculating new state: $h_t = o_t * \tanh\ (c_t)$


h_t
C_{t-1}
C_t
σ
σ
σ
h_{t-1}
h_t
X_t
0
1
2
3
Inputs:
X_t Input vector
C_{t-1} Memory from previous block
h_{t-1} Output of previous block
outputs:
C_t Memory from current block
h_t Output of current block
Nonlinearities:
σ Sigmoid
Hyperbolic tangent
Bias:
Vector operations:
× Element-wise multiplication
+ Element-wise Summation / Concatenation

As it is seen in the equations, each gate has various arrangements of weights. In the last equation, the input gate and intermediate cell state are included with the old cell state and the forget gate. The output of this is used to calculate the new state. This propelled cell with four connecting layers of only one tanh layer in RNN, make LSTM ideal for sequence prediction.

Network Architecture: - We are using the sequential APIs of Keras. The general architecture is:

<u>LSTM layer: -</u> The LSTM layer is the internal one, and all the gates, cited at the starting are carried out with the implementation of Keras, with a default activation of hard-sigmoid.

<u>Dropout layer:</u> - This is used before the dense layer. For Keras, after any hidden layer a dropout can be added, but we add it after the LSTM.

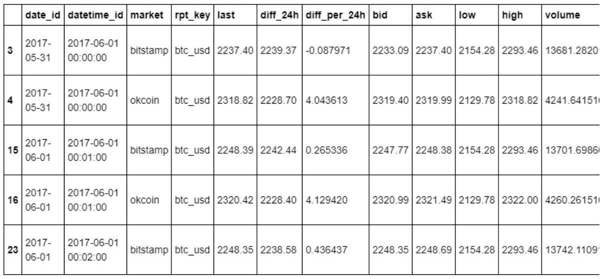
<u>Dense layer:</u> - This layer is a connected layer.

<u>Activation layer</u>: - Because we are fixing a regression problem, the last layer ought to supply the linear combination of the activations of the previous layer with weight vectors. Therefore, this activation is a linear one. As an alternative, it could be passed as a parameter to the previous dense layer.

IV. RESULT AND ANALYSIS

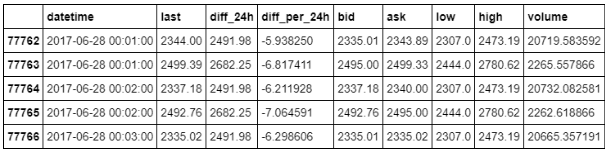
It is observed that if the training set is huge then it is more prone to overfitting problem. Neural networks provided a better understanding of bitcoin and LSTM architecture. The LSTM works efficiently and provides us with the most accurate results.

Below is the snapshot of a few records from actual dataset to analyze the final price of bitcoins.



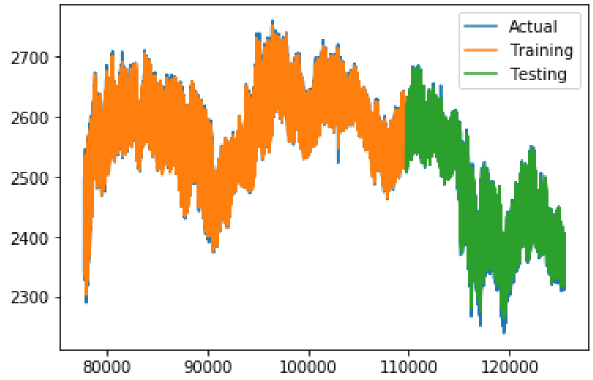
	date_id	datetime_id	market	rpt_key	last	diff_24h	diff_per_24h	bid	ask	low	high	volume
3	2017-05-31	2017-06-01 00:00:00	bitstamp	btc_usd	2237.40	2239.37	-0.087971	2233.09	2237.40	2154.28	2293.46	13681.2820
4	2017-05-31	2017-06-01 00:00:00	okcoin	btc_usd	2318.82	2228.70	4.043613	2319.40	2319.99	2129.78	2318.82	4241.64151
15	2017-06-01	2017-06-01 00:01:00	bitstamp	btc_usd	2248.39	2242.44	0.265336	2247.77	2248.38	2154.28	2293.46	13701.6986
16	2017-06-01	2017-06-01 00:01:00	okcoin	btc_usd	2320.42	2228.40	4.129420	2320.99	2321.49	2129.78	2322.00	4260.26151
23	2017-06-01	2017-06-01 00:02:00	bitstamp	btc_usd	2248.35	2238.58	0.436437	2248.35	2248.69	2154.28	2293.46	13742.1109

Intermediate processed data set results is shown i.e. after processing all the dates greater than the mentioned date.



	datetime	last	diff_24h	diff_per_24h	bid	ask	low	high	volume
77762	2017-06-28 00:01:00	2344.00	2491.98	-5.938250	2335.01	2343.89	2307.0	2473.19	20719.583592
77763	2017-06-28 00:01:00	2499.39	2682.25	-6.817411	2495.00	2499.33	2444.0	2780.62	2265.557866
77764	2017-06-28 00:02:00	2337.18	2491.98	-6.211928	2337.18	2340.00	2307.0	2473.19	20732.082581
77765	2017-06-28 00:02:00	2492.76	2682.25	-7.064591	2492.76	2495.00	2444.0	2780.62	2262.618866
77766	2017-06-28 00:03:00	2335.02	2491.98	-6.298606	2335.01	2335.02	2307.0	2473.19	20665.357191

The final result is shown in the form of a graph. The graph consists of the actual dataset, test dataset and the training dataset. It is clearly shown in the graph that the test dataset predicts the output very close to the actual dataset. Therefore, the prediction obtained is more accurate.



V. CONCLUSION

Although various theories and algorithms have been developed for the prediction of the price of bitcoins, most of them have been proved that they needed to be reconsidered for reducing problems of overfitting and errors resulting from high sized datasets. The value of bitcoin in the future can be predicted using the LSTM algorithm. Because of the usage of this algorithm, we can save a large amount of data and predict the most accurate results.

VI. REFERENCES

[1] S. Nakamoto (2008) "Bitcoin: A peer-to-peer electronic cash system," URL:https://bitcoin.org/bitcoin.pdf

[2] G. Heleman and M. Rauchs (2018) "Global cryptocurrency benchmarking study," 10th International Conference on Information Technology and Electrical Engineering (ICITEE)

[3] G. Neil and H. Halaburda (2016) "Can we predict the winner in a market with network effects? Competition in cryptocurrency market," Games, vol.7 no.3, pp. 16.

[4] Madan, S. Saluja and A. Zhao (2015) "Automated Bitcoin Trading via Machine Learning Algorithms," URL: http//cs229.stanford.edu/proj2015/,vol. 20, pp. 1–5.

[5] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton,J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux (2013) "API design for machine learning software: experiences from the scikitlearn project," arXiv preprint arXiv:1309.0238.

[6] C. François (2015) "Keras: Deep learning library for theano and tensorflow," URL: https://keras. io/k.

[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G S. Corrado, A. Davis, J. Dean, M.Devin, R.Monga, S.Moore, D. Murray, C.Olah, M. Schuster, J. Shlens, B.Steiner, I. Sutskever, K.Talwar, P.Tucker, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng (2016) "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," arXiv preprint arXiv:1603.04467.

[8] Shah, Devavrat and Kang Zhang (2014) "Bayesian regression and Bitcoin." arXiv preprint arXiv:1410.1231.

[9] G. P. Dwyer (2015) "The economics of Bitcoin and similar private digital currencies," Journal of Financial Stability, vol. 17, pp. 81–91

[10] M. Iwamura, Y. Kitamura, and T. Matsumoto (2014) "Is bitcoin the only cryptocurrency in the town? economics of cryptocurrency and friedrich a. hayek," SSRN Electronic Journal.

[11] F.Sheta, S.E.M.Ahmed, and H.Faris (2015) "A comparison between regression, artificial neural networks and support vector machines for predicting stock market index," Soft Computing, vol. 7, p. 8.

[12] H. Jang and J. Lee (2017) "An Empirical Study on Modeling and Prediction of Bitcoin Prices with Bayesian Neural Networks Based on Blockchain Information," IEEE Access, vol. 6, pp. 5427–5437.

[13] S. Hochreiter and J. Schmidhuber (1997) "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780.

[14] Huisu Jang and Jaewook Lee (2017) "An Empirical Study on Modelling and Prediction of Bitcoin Prices with Bayesian Neural Networks based on Blockchain Information," in IEEE Early Access Articles, vol. 99, pp. 1-1.

[15] M. Daniela and A. BUTOI (2013) "Data mining on Romanian stock market using neural networks for price prediction". informatica Economica, 17.

ACKNOWLEDGEMENT

1. **Reshma Sundari Gadey** is pursuing B. Tech in the stream of Computer Science and Engineering in Institute of Aeronautical Engineering, Dundigal, Hyderabad, Telangana, India.
2. **Dr. R Obulakonda Reddy** is an Assistant Professor in the Department of Computer Science in Institute of Aeronautical Engineering, Dundigal, Hyderabad, Telangana, India.
3. **Nikita Thakur** is pursuing B. Tech in the stream of Computer Science and Engineering in Institute of Aeronautical Engineering, Dundigal, Hyderabad, Telangana, India.
4. **Naveen Charan** is pursuing B. Tech in the stream of Computer Science and Engineering in Institute of Aeronautical Engineering, Dundigal, Hyderabad, Telangana, India.