PROJECT REPORT
(Project Term January-March 2022)

BITCOIN PRICE PREDICTION

Submitted by
Devansh Agarwal
11908637

Project Group Number- KM015
Course Code- INT247

Under the Guidance of Sir Sagar Pande

School of Computer Science and Engineering

DECLARATION

I hereby declare that the project work entitled “Bitcoin Price Prediction” is an authentic record of my own work carried out as requirements of Project for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Sir Sagar Pande, during January to March 2022. All the information furnished in this project report is based on my own intensive work and is genuine.

Devansh Agarwal
11908637 March 29, 2022

ACKNOWLEDGEMENT

It is with my immense gratitude that I acknowledge the support and help of my Professor, Mr. Sagar Pande, who has always encouraged me into this research. Without his continuous guidance and persistent help, this project would not have been a success for me. I am grateful to the Lovely Professional University, Punjab and the department of Computer Science without which this project would have not been an achievement. I also thank my family and friends, for their endless love and support throughout my life.

TABLE OF CONTENTS

ABSTRACT

In this project, we proposed to predict the Bitcoin price accurately taking into consideration various parameters that affect the Bitcoin value. By gathering information from different reference papers and applying in real time, I found the advantages and disadvantages of bitcoin price prediction.

Each and every paper has its own set of methodologies of bitcoin price prediction. Many papers have accurate price but some other don't, but the time complexity is higher in those predictions, so to reduce the time complexity here in this paper we use a model linked to artificial intelligence named LSTM. The other papers used different models which do not have a great time management, but in LSTM finding of the results from a larger database is quick and fast. So, for this purpose we draw a comparison between other models and LSTM model, this survey paper helps the upcoming researchers to make an impact in their papers. The process happens in the paper is first moment of the research, we aim to understand and find daily trends in the Bitcoin market while gaining insight into optimal features surrounding Bitcoin price. Our data set consists of various features relating to the Bitcoin price and payment network over the course of every years, recorded daily. By preprocessing the dataset, we apply some data mining techniques to reduce the noise of data. Then the second moment of our research, using the available information, we will predict the sign of the daily price change with highest possible accuracy.

INTRODUCTION

The Main objective of this project is to predict the bitcoin using Machine Learning Algorithms. The model is based on long short-term memory (LSTM) recurrent neural networks. In all cases, we build investment portfolios based on the predictions and we compare their performance in terms of return on investment.

Machine learning and AI-assisted trading have attracted growing interest for the past few years. Here, we use this approach to test the hypothesis that the inefficiency of the cryptocurrency market can be exploited to generate abnormal profits. We analyze daily data for 1, 681 cryptocurrencies for the period between Sept 2014 and Mar 2022. We show that simple trading strategies assisted by state-of-the-art machine learning algorithms outperform standard benchmarks. Our results show that non-trivial, but ultimately simple, algorithmic mechanisms can help anticipate the short-term evolution of the cryptocurrency market.

The popularity of cryptocurrencies has skyrocketed in 2017 due to several consecutive months of super-exponential growth of their market capitalization. Today, there are more than 1, 500 actively traded cryptocurrencies capitalizing over $300 billion, with a peak of the market capitalization totaling more than $800 billion in Jan. 2018. Between 2.9 and 5.8 million privates as well as institutional investors are in the different transaction networks, according to a recent survey, and access to the market has become easier over time. Major cryptocurrencies can be bought using fiat currency in a number of online exchanges and then be used in their turn to buy less popular cryptocurrencies. The volume of daily exchanges is currently superior to $15 billion. Since 2017, over 170 hedge funds specialized in cryptocurrencies have emerged and bitcoin futures have been launched to address institutional demand for trading and hedging Bitcoin.

HISTORY

With the appearance of Bitcoin approx. 24 years ago the global economist, albeit in small numbers, is flexible and responsive. Bitcoin introduced itself as a program that solved the Double Spend problem (Nakamoto & Shah, 2017) , a preferred issue with Digital Cash systems. However, the impact in the coming years is 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.

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 its 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 twenty-four 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%.

DATA PREPROCESSING

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. 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.

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.

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

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**.**

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)

ARCHITECTURE OF LSTM

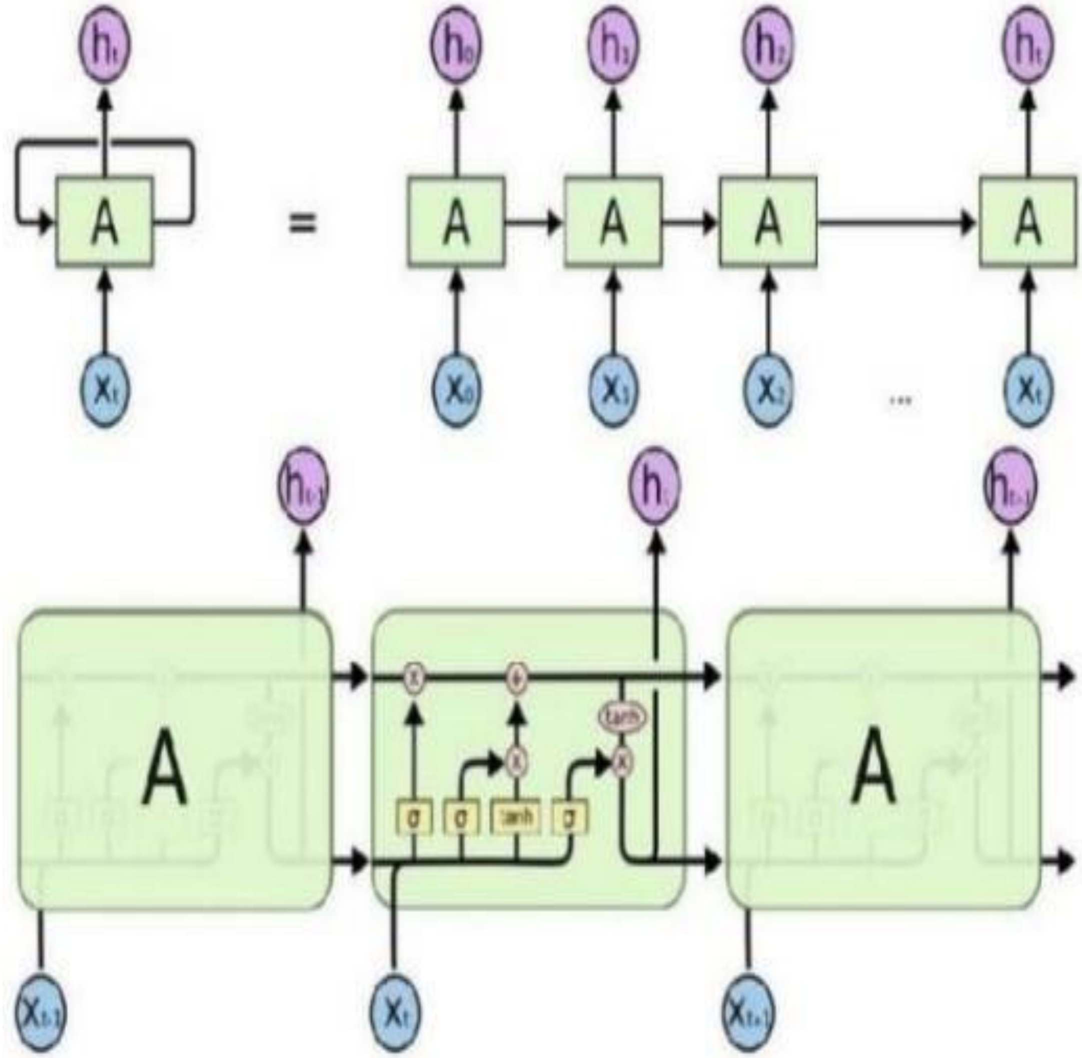
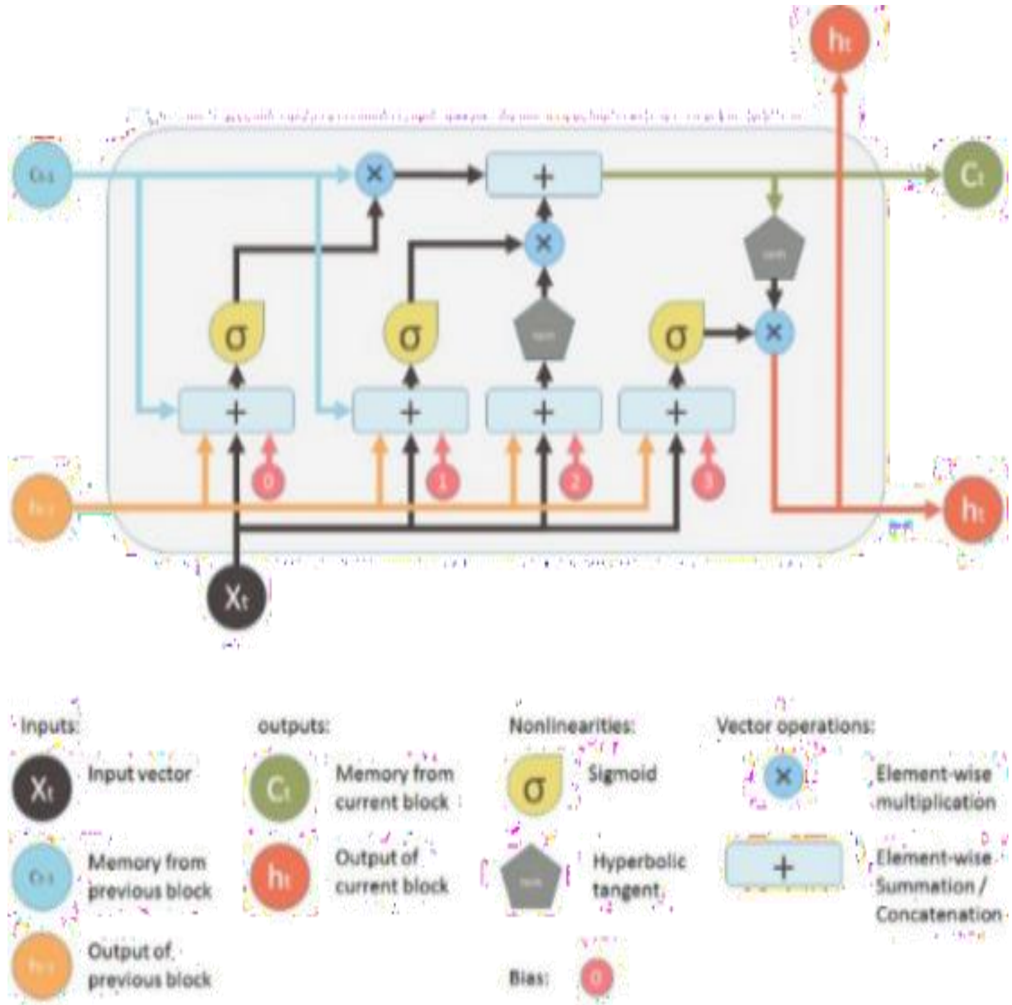


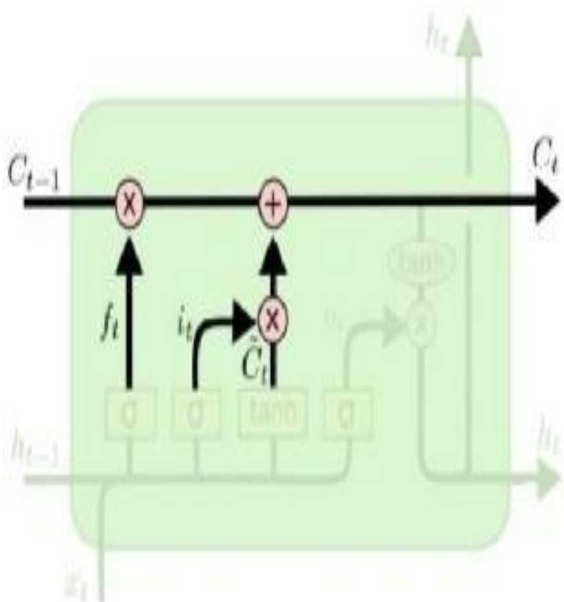
Figure 3:



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

Figure 4:

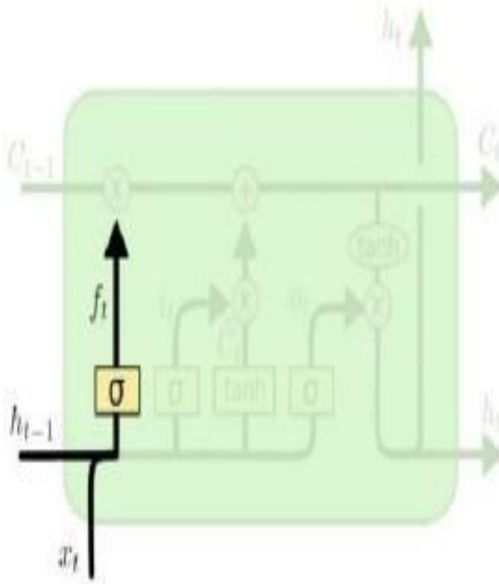
The current state



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

Figure 5:

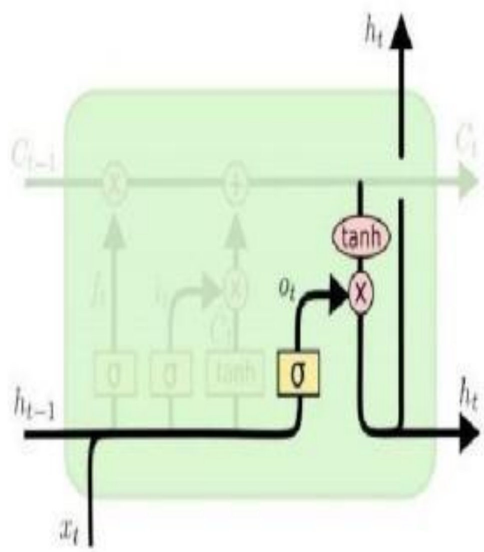
Forget gate layer



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

Figure 6:

Output layer

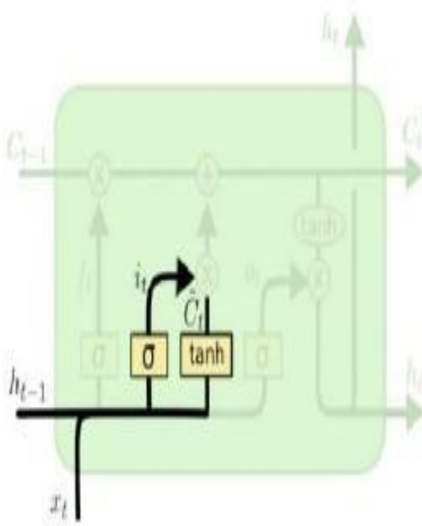


$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$

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

Figure 7:

Input gate layer



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 8:

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.

Existing System:

In the existing system, we analyzed stock markets prediction, which suggests that these methods could be effective also in predicting cryptocurrencies prices. However, the application of machine learning algorithms to the cryptocurrency market has been limited so far to the analysis of Bitcoin prices, using random forests, Bayesian neural network, long short-term memory neural network? and other algorithms. These studies were able to anticipate, to different degrees, the price fluctuations of Bitcoin, and revealed that best results were achieved by neural network-based algorithms. Deep reinforcement learning was shown to beat the uniform buy and hold strategy in predicting the prices of 12 cryptocurrencies over one-year period.

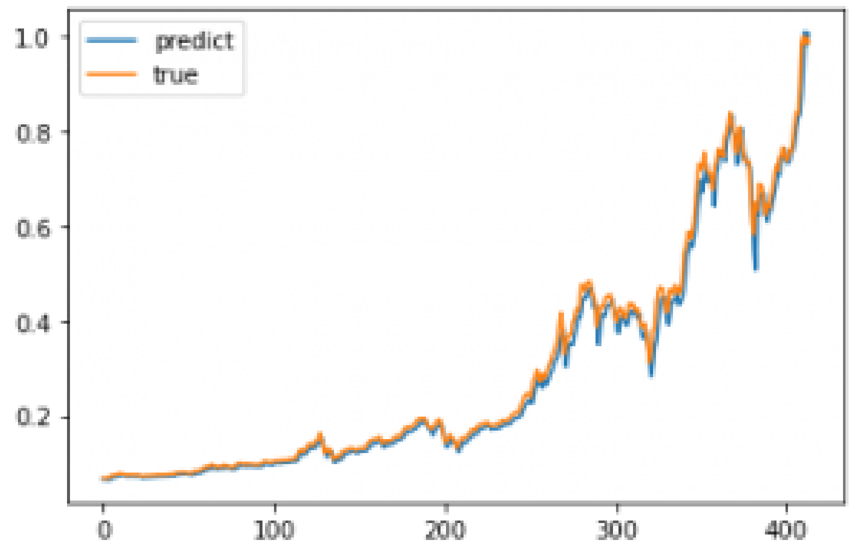
Disadvantage:

1. Other attempts to use machine learning to predict the prices of cryptocurrencies other than Bitcoin come from non-academic sources.

2. Most of these analyses focused on a limited number of currencies and did not provide benchmark comparisons for their results.

Proposed System:

Here, we test the performance of three models in predicting daily cryptocurrency prices for 1,681 currencies. Two of the models are based on gradient boosting decision trees and one is based on long short-term memory (LSTM) recurrent neural networks. In all cases, we build investment portfolios based on the predictions and we compare their performance in terms of return on investment. We find that all of the three models perform better than a baseline a simple moving average? model? where a currency's price is predicted as the average price across the preceding days, and that the method based on long short-term memory recurrent neural networks systematically yields the best return on investment.

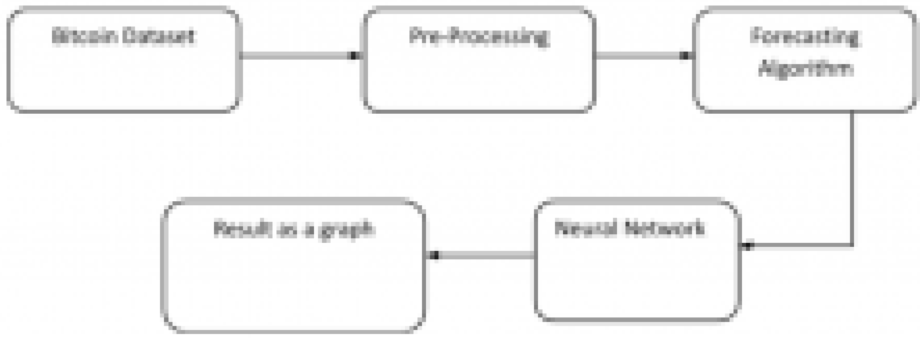


Bitcoin

Price Prediction using Machine Learning Python 7

Advantage:

1. We present and compare the results obtained with the three forecasting algorithms and the baseline method.
2. We predict the price of the currencies at day for all included between Jan, 1st 2016 and Apr 24th, 2018.
3. The analysis considers all currencies whose age is larger than 50 days since their first appearance and whose volume is larger than $100000.
4. To discount for the effect of the overall market movement (i.e., market growth, for most of the considered period), we consider cryptocurrencies prices expressed in Bitcoin.


Bitcoin Dataset
Pre-Processing
Forecasting Algorithm
Neural Network
Result as a graph

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.

Desktop/ML/ | bitcoinprice prediction - Jupyter | Bitcoin-Price-Prediction/bitcoinp

localhost:8888/notebooks/Desktop/ML/bitcoinprice%20prediction.ipynb

jupyter bitcoinprice prediction Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
```

In [2]:
```
data = pd.read_csv('BTC-USD.csv', date_parser = True)
data.tail()
```

Out[2]:

	Date	Open	High	Low	Close	Adj Close	Volume
2747	2022-03-26	44349.859375	44735.996094	44166.273438	44500.828125	44500.828125	16950455995
2748	2022-03-27	44505.355469	46827.546875	44437.292969	46820.492188	46820.492188	28160889722
2749	2022-03-28	46821.851563	48086.835938	46690.203125	47128.003906	47128.003906	36362175703
2750	2022-03-29	47100.437500	48022.289063	47100.437500	47465.730469	47465.730469	31397059069
2751	2022-03-30	47375.777344	47544.847656	46759.917969	47127.250000	47127.250000	31972411392

In [3]:
```
data_training = data[data['Date']< '2022-03-01'].copy()
data_training
```

Out[3]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100

Desktop/ML/ | bitcoinprice prediction - Jupyter | Bitcoin-Price-Prediction/bitcoinp

localhost:8888/notebooks/Desktop/ML/bitcoinprice%20prediction.ipynb

jupyter bitcoinprice prediction Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

2749	2022-03-28	46821.851563	48086.835938	46690.203125	47128.003906	47128.003906	36362175703
2750	2022-03-29	47100.437500	48022.289063	47100.437500	47465.730469	47465.730469	31397059069
2751	2022-03-30	47375.777344	47544.847656	46759.917969	47127.250000	47127.250000	31972411392

In [3]:
```
data_training = data[data['Date']< '2022-03-01'].copy()
data_training
```

Out[3]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100
...	...	...	...	...	...	...	...
2717	2022-02-24	37278.566406	38968.839844	34459.218750	38332.609375	38332.609375	46383802093
2718	2022-02-25	38333.746094	39630.324219	38111.343750	39214.218750	39214.218750	26545599159
2719	2022-02-26	39213.082031	40005.347656	38702.535156	39105.148438	39105.148438	17467554129
2720	2022-02-27	39098.699219	39778.941406	37268.976563	37709.785156	37709.785156	23450127612
2721	2022-02-28	37706.000000	43760.457031	37518.214844	43193.234375	43193.234375	35690014104

2722 rows × 7 columns

In [4]:
```
data_test = data[data['Date']> '2022-03-01'].copy()
data_test
```

Desktop/ML/ | bitcoinprice prediction - Jupyter | Bitcoin-Price-Prediction/bitcoinp

localhost:8888/notebooks/Desktop/ML/bitcoinprice%20prediction.ipynb

jupyter bitcoinprice prediction Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

2722 rows × 7 columns

In [4]:

```
data_test = data[data['Date']> '2022-03-01'].copy()
data_test
```

Out[4]:

	Date	Open	High	Low	Close	Adj Close	Volume
2723	2022-03-02	44357.617188	45077.578125	43432.851563	43924.117188	43924.117188	29183112630
2724	2022-03-03	43925.195313	44021.578125	41914.750000	42451.789063	42451.789063	24967782593
2725	2022-03-04	42458.140625	42479.613281	38805.847656	39137.605469	39137.605469	28516271427
2726	2022-03-05	39148.449219	39566.335938	38777.035156	39400.585938	39400.585938	16975917450
2727	2022-03-06	39404.199219	39640.175781	38211.648438	38419.984375	38419.984375	19745229902
2728	2022-03-07	38429.304688	39430.226563	37260.203125	38062.039063	38062.039063	28546143503
2729	2022-03-08	38059.902344	39304.441406	37957.386719	38737.269531	38737.269531	25776583476
2730	2022-03-09	38742.816406	42465.671875	38706.093750	41982.925781	41982.925781	32284121034
2731	2022-03-10	41974.070313	42004.726563	38832.941406	39437.460938	39437.460938	31078064711
2732	2022-03-11	39439.968750	40081.679688	38347.433594	38794.972656	38794.972656	26364890465
2733	2022-03-12	38794.464844	39308.597656	38772.535156	38904.011719	38904.011719	14616450657
2734	2022-03-13	38884.726563	39209.351563	37728.144531	37849.664063	37849.664063	17300745310
2735	2022-03-14	37846.316406	39742.500000	37680.734375	39666.753906	39666.753906	24322159070
2736	2022-03-15	39664.250000	39794.628906	38310.210938	39338.785156	39338.785156	23934000868
2737	2022-03-16	39335.570313	41465.453125	39022.347656	41143.929688	41143.929688	39616916192
2738	2022-03-17	41140.843750	41287.535156	40662.871094	40951.378906	40951.378906	22009601093
2739	2022-03-18	40944.839844	42195.746094	40302.398438	41801.156250	41801.156250	34421564942

Desktop/ML/ | bitcoinprice prediction - Jupyter | Bitcoin-Price-Prediction/bitcoinp

localhost:8888/notebooks/Desktop/ML/bitcoinprice%20prediction.ipynb

jupyter bitcoinprice prediction Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

2730	2022-03-09	38742.816406	42465.671875	38706.093750	41982.925781	41982.925781	32284121034
2731	2022-03-10	41974.070313	42004.726563	38832.941406	39437.460938	39437.460938	31078064711
2732	2022-03-11	39439.968750	40081.679688	38347.433594	38794.972656	38794.972656	26364890465
2733	2022-03-12	38794.464844	39308.597656	38772.535156	38904.011719	38904.011719	14616450657
2734	2022-03-13	38884.726563	39209.351563	37728.144531	37849.664063	37849.664063	17300745310
2735	2022-03-14	37846.316406	39742.500000	37680.734375	39666.753906	39666.753906	24322159070
2736	2022-03-15	39664.250000	39794.628906	38310.210938	39338.785156	39338.785156	23934000868
2737	2022-03-16	39335.570313	41465.453125	39022.347656	41143.929688	41143.929688	39616916192
2738	2022-03-17	41140.843750	41287.535156	40662.871094	40951.378906	40951.378906	22009601093
2739	2022-03-18	40944.839844	42195.746094	40302.398438	41801.156250	41801.156250	34421564942
2740	2022-03-19	41794.648438	42316.554688	41602.667969	42190.652344	42190.652344	19664853187
2741	2022-03-20	42191.406250	42241.164063	41004.757813	41247.824219	41247.824219	20127946682
2742	2022-03-21	41246.132813	41454.410156	40668.042969	41077.996094	41077.996094	24615543271
2743	2022-03-22	41074.105469	43124.707031	40948.281250	42358.808594	42358.808594	32004652376
2744	2022-03-23	42364.378906	42893.507813	41877.507813	42892.957031	42892.957031	25242943069
2745	2022-03-24	42886.652344	44131.855469	42726.164063	43960.933594	43960.933594	31042992291
2746	2022-03-25	43964.546875	44999.492188	43706.285156	44348.730469	44348.730469	30574413034
2747	2022-03-26	44349.859375	44735.996094	44166.273438	44500.828125	44500.828125	16950455995
2748	2022-03-27	44505.355469	46827.546875	44437.292969	46820.492188	46820.492188	28160889722
2749	2022-03-28	46821.851563	48086.835938	46690.203125	47128.003906	47128.003906	36362175703
2750	2022-03-29	47100.437500	48022.289063	47100.437500	47465.730469	47465.730469	31397059069
2751	2022-03-30	47375.777344	47544.847656	46759.917969	47127.250000	47127.250000	31972411392

2744	2022-03-23	42364.378906	42893.507813	41877.507813	42892.957031	42892.957031	25242943069
2745	2022-03-24	42886.652344	44131.855469	42726.164063	43960.933594	43960.933594	31042992291
2746	2022-03-25	43964.546875	44999.492188	43706.285156	44348.730469	44348.730469	30574413034
2747	2022-03-26	44349.859375	44735.996094	44166.273438	44500.828125	44500.828125	16950455995
2748	2022-03-27	44505.355469	46827.546875	44437.292969	46820.492188	46820.492188	28160889722
2749	2022-03-28	46821.851563	48086.835938	46690.203125	47128.003906	47128.003906	36362175703
2750	2022-03-29	47100.437500	48022.289063	47100.437500	47465.730469	47465.730469	31397059069
2751	2022-03-30	47375.777344	47544.847656	46759.917969	47127.250000	47127.250000	31972411392

In [5]:

```
training_data = data_training.drop(['Date', 'Adj Close'], axis = 1)
training_data.head()
```

Out[5]:

	Open	High	Low	Close	Volume
0	465.864014	468.174011	452.421997	457.334015	21056800
1	456.859985	456.859985	413.104004	424.440002	34483200
2	424.102997	427.834991	384.532013	394.795990	37919700
3	394.673004	423.295990	389.882996	408.903992	36863600
4	408.084991	412.425995	393.181000	398.821014	26580100

In [8]:

```
scaler = MinMaxScaler()
training_data = scaler.fit_transform(training_data)
training_data
```

Out[8]:

```
array([[4.28907290e-03, 3.73944128e-03, 4.24270741e-03, 4.14358659e-03,
        4.31449241e-05],
       [4.15542811e-03, 3.57446063e-03, 3.64887469e-03, 3.65546320e-03,
```

```
54/54 [==============================] - 12s 220ms/step - loss: 0.0026
Epoch 17/20
54/54 [==============================] - 12s 215ms/step - loss: 0.0022
Epoch 18/20
54/54 [==============================] - 12s 218ms/step - loss: 0.0024
Epoch 19/20
54/54 [==============================] - 12s 226ms/step - loss: 0.0025
Epoch 20/20
54/54 [==============================] - 12s 225ms/step - loss: 0.0021
```

Out[19]: `<keras.callbacks.History at 0x21ac568cf70>`

In [20]:

```
#Test Dataset
past_60_days = data_training.tail(60)
df= past_60_days.append(data_test, ignore_index = True)
df = df.drop(['Date', 'Adj Close'], axis = 1)
df.head()
```

Out[20]:

	Open	High	Low	Close	Volume
0	47169.371094	48472.527344	45819.953125	46306.445313	36974172400
1	46311.746094	47827.312500	46288.484375	47686.812500	24582667004
2	47680.925781	47881.406250	46856.937500	47345.218750	27951569547
3	47343.542969	47510.726563	45835.964844	46458.117188	33071628362
4	46458.851563	47406.546875	45752.464844	45897.574219	42494677905

In [21]:

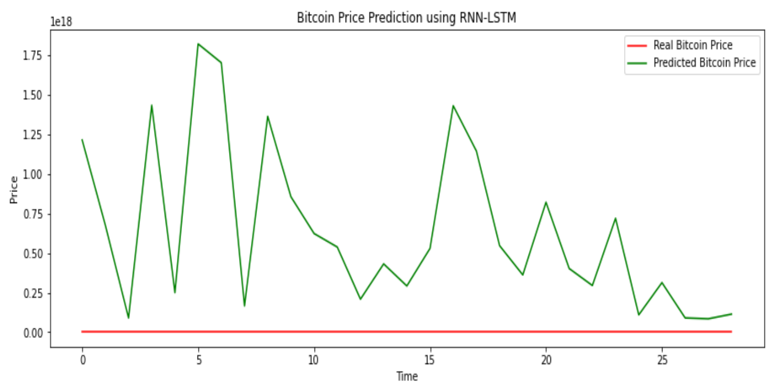
```
inputs = scaler.transform(df)
inputs
```

Out[21]:

```
array([[4.71693711e+04, 4.84725273e+04, 4.58199531e+04, 4.63064453e+04,
        3.69741724e+10],
```

```
8.55903669e+08, 8.58904573e+08, 9.03610408e+08, 9.08986812e+08,
9.14300569e+08])
```

In [30]:
```
plt.figure(figsize=(14,5))
plt.plot(Y_test, color = 'red', label = 'Real Bitcoin Price')
plt.plot(Y_pred, color = 'green', label = 'Predicted Bitcoin Price')
plt.title('Bitcoin Price Prediction using RNN-LSTM')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()
```



In [ ]:

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

Hardware and Software Requirements:

Hardware:

1.OS: - Windows 7,8 or 10 (32 or 64 bit)

2.RAM: -4GB

Software:

1.Python IDLE

2.Anaconda

3. Jupiter Notebook

4. 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.

References

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

- Jang, H. & Lee, J. 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,”. 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*.

- Santhosh, &., S. G. &”. A Relative Study on Bitcoin Mining. *Relative Study on Bitcoin Mining”, Imperial Journal of Interdisciplinary Research*.

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