# Automated Blockchain Transactions with Parallel model comparison for best price Forecast of Bitcoin

## Abstract:

In this project, I attempt to apply machine-learning algorithms to predict Bitcoin price and thus automating the flow of transaction (like buying, selling or long, short) of Bitcoins based on the predicted deviations is the forecasted price. For the first phase of the investigation, I aimed to understand and better identify daily trends in the Bitcoin market while gaining insight into the most optimal methods and techniques to successfully predict Bitcoin price. At the same time, a comparison has also been drawn to gauge the performance of the most popular machine learning algorithms which can be used to predict the Bitcoin prices. Three prominent models were tested for this purpose: Bagging Algorithm (Random Forest), Boosting Algorithm (XGBoost) and Deep Learning Algorithm (RNN with LSTM). The best results are observed in the case of Deep Learning algorithm with a maximum accuracy of 99.23%, which is significantly higher than most of the prevalent models today. In the second phase, we developed an automated signal mechanism which is capable of producing a signal whether to short or long Bitcoins on any given day. The signals have been classified into three categories: Weak, Moderate and Strong.

# 1. Introduction

## 1.1 Blockchain

The concept of Blockchain first came into light in October 2008, as part of a proposal for Bitcoin, with the aim to create peer-to-peer money without intermediaries (or banks). Bitcoin introduced a novel solution to one of the most primitive problems of trust. The proposed blockchain technology enables us to trust the outputs of the system without trusting any intermediary within it. People and institutions who do not know or cannot trust each other, living in different countries, are subject to different jurisdictions, and who have no legally binding agreements with each other, can now interact over the Internet without the need for intermediaries like banks, Internet platforms, or other types of clearing institutions.

## 1.2 Price Prediction of Bitcoin

The Bitcoin market's financial analogue is, of course, a stock market. To maximize financial reward, the field of stock market prediction has grown over the past decades and has more recently exploded with the advent of high-frequency, low-latency trading hardware coupled with robust machine learning algorithms. Thus, it cannot go unnoticed that this prediction methodology is replicated in the world of Bitcoin, as the network gains greater liquidity and more people start engaging in the system to earn profits. To do so, I feel it is necessary to leverage machine learning technology to predict the price of Bitcoin. All the models that are presented have been implemented here use a time series structure for prediction.

# 2. Methodology

## 2.1. Data Collection:

There are several resources which one can use to obtain or extract historical Bitcoin price data. While some of these resources allow us to manually download data files like CSV manually, others provide an API that one can hook up to their code. The data used in training and testing of all our models have been taken from Kaggle's datasets for Bitcoin Historical Data [1].

## 2.2. Data Manipulation:

The data that was obtained was in the following form (after data cleaning):

| Timestamp | Open | High | Low | Close | Volume_(BTC) | Volume_(Currency) | Weighted_Price |
|---|---|---|---|---|---|---|---|
| 1325317920 | 4.39 | 4.39 | 4.39 | 4.39 | 0.455581 | 2.000000 | 4.390000 |
| 1325346600 | 4.39 | 4.39 | 4.39 | 4.39 | 48.000000 | 210.720000 | 4.390000 |
| 1325350740 | 4.50 | 4.57 | 4.50 | 4.57 | 37.862297 | 171.380338 | 4.526411 |
| 1325350800 | 4.58 | 4.58 | 4.58 | 4.58 | 9.000000 | 41.220000 | 4.580000 |
| 1325391360 | 4.58 | 4.58 | 4.58 | 4.58 | 1.502000 | 6.879160 | 4.580000 |

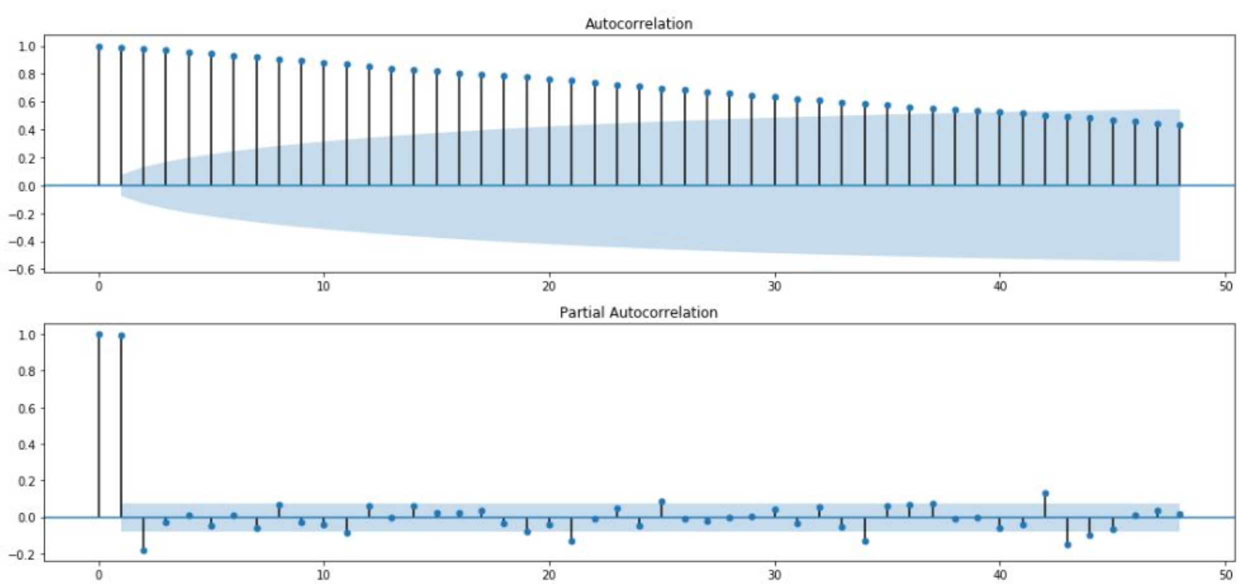
To make proper predictions, we require to transform the data to get the average price grouped by the day or date and to see usual DateTime format (not a timestamp as above). Moreover, we also require to split our dataset into training and testing (or validation) sets for proper modelling. Here, I have trained the model on the data from January 1, 2016, until August 21, 2017, and to test the model on the data from August 21, 2017, until October 20, 2017. The earlier dates have been ignored as those will not have much impact on the current prices in our time series analysis.

## 2.3. Exploratory Data Analysis

We need to estimate some parameters of the data because this can be useful in further model designing. The first aspect which we need to take care of while working with time series data is to check whether the data is stationary or not. This basically means that we need to check if our data is influenced by factors like trend or seasonality.

Following this, I have also performed a seasonal decomposition of the data to visualize as well as estimate the popular trend and seasonality in the data. In the figure shown below, you can easily observe the actual price movements on the plot and also the trend and seasonality in our data.


The next thing that I did is the examination of the autocorrelation. It is nothing but the similarity between different observations which can be estimated as a function of the time gap between them. It is important to find repeating patterns in any data especially time series or historical data.



## 2.4. Data Preparation

After analysing the data, we need to prepare the dataset in accordance with the requirements of the model, as well as we need to split the dataset into train and test sets for proper modelling. Thus, I defined a function which creates X inputs and Y labels for the model. While performing sequential forecasting, we predict future values based on some previous values as well as some current values. Thus, the Y label is the value from the future point of time while the X are the one or several values from the past or previous set. The magnitude or degree of these values can be set by tuning the parameter 'look_back' in the function 'create_lookback'. For this purpose, I have set it to 1, which means that we are predicting any current value say 't' based on some previous value (t-1).

After this, the final data preparation is performed:

1. Reshape the training and testing data as required by the model.
2. Scaling the dataset by using a MinMaxScaler function because models like LSTM are scale sensitive.
3. Apply our 'create_lookback' function.

## 2.5. Training

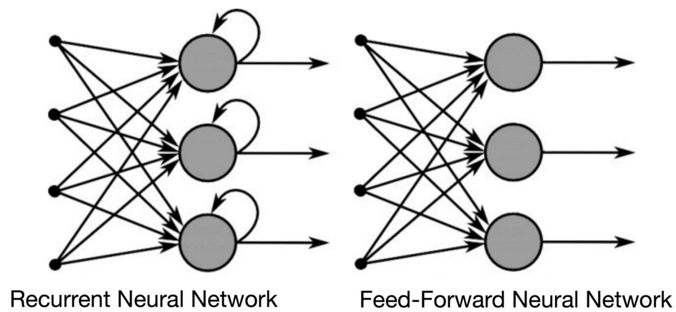
For training we target the three most popular types of algorithms:

- **Bagging Algorithm: RandomForest**
  - I used RandomForestRegressor as they are often accurate, do not require feature scaling, categorical feature encoding, and need little parameter tuning. They can also be more interpretable than other complex models such as neural networks.

- **Boosting Algorithm: XGBoost**
  - XGBoost (eXtreme Gradient Boosting) is an advanced implementation of gradient boosting algorithm. The reason for choosing XGBoost is its relative advantages like regularized boosting, high flexibility, efficient tree pruning and built-in cross-validation. For XGBoost, the hyperparameter tuning was accomplished using a simple GridSearchCV function [2]

- **Deep Learning Algorithm: Recurrent Neural Network (RNN) using LSTM**

- A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. [3]. Let's focus more on this algorithm as it achieved the best result in the study.
  Since we are using a time series dataset, it is not viable to use a feedforward-only neural network as tomorrow's Bitcoin price is most correlated with today's, not a month ago's. An RNN shows temporal dynamic behaviour for a time sequence and it can use its internal state to process sequences [4]. In practice, this can be achieved with LSTMs and GRUs layers.
  Here you can see the difference between a regular feedforward-only neural network and a recurrent neural network (RNN)




To be able to create a program that trains on the historical Bitcoin prices and forecast future prices, we need to complete several tasks as follows:

- Data Cleaning and preparation (as stated above)
- Building an RNN with LSTM (creating a Sequential model with two LSTM and two Dense layers)
- Training this RNN model.
- Predicting future Bitcoin prices.

## 2.6. Automated Bitcoin Transaction Signals

After achieving significantly higher accuracy while prediction future Bitcoin prices, we can now produce signals for the transactions using these predictions. For this, we first need to detrend the prediction. After detrending the price data, we can now signal out 6 different messages stating whether to sell or buy and at which intensity:

- Buy (High Intensity): if predicted price >= Current Price + 1.5*(Standard Deviation)
- Buy (Moderate Intensity): if Current Price + 1.5*(Standard Deviation) > predicted price >= Current + 0.5*(Standard Deviation)
- Buy (Low Intensity): if  Current + Standard Deviation > predicted price >= Current

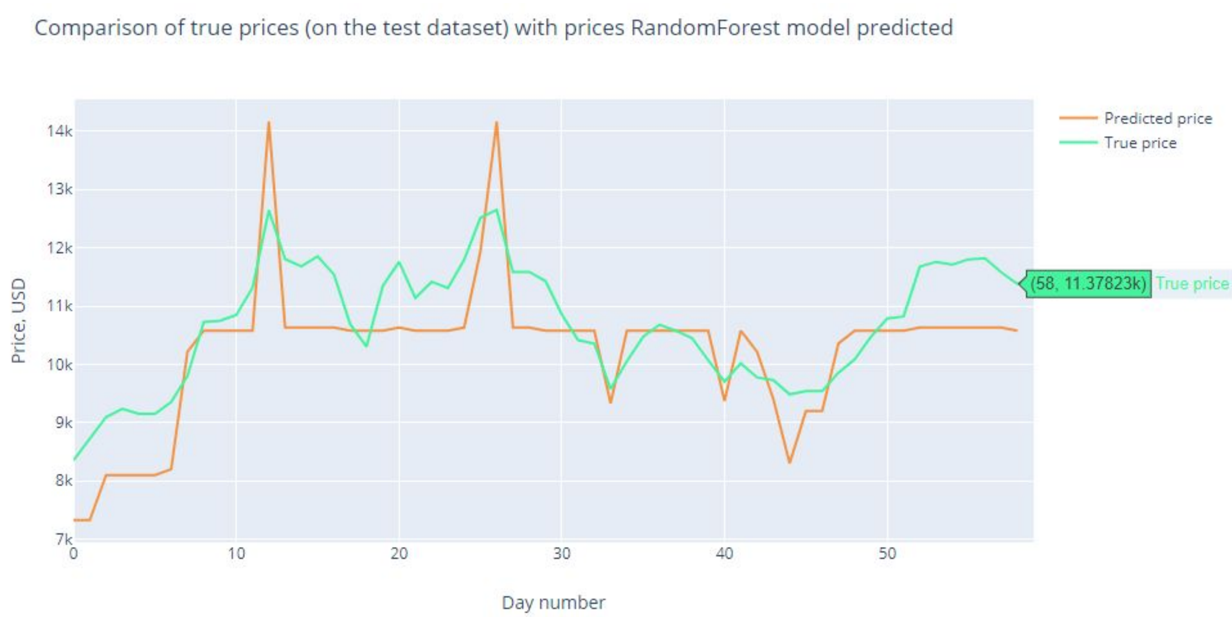
● Sell (High Intensity): if predicted price =< Current - 1.5*(Standard Deviation)
● Sell (Moderate Intensity): if Current - 1.5*(Standard Deviation) < predicted price =< Current - 0.5*(Standard Deviation)
● Sell (Low Intensity): if Current - Standard Deviation < predicted price =< Current Price

# 3. Performance & Results:

As stated earlier, the performance of the Deep Learning model (RNN with LSTM) turned out to be the best with a maximum accuracy of 99.23%. The model consisted of a sequential network with two LSTM layer and two Dense layers.
The poorest performance is observed in the case of Random Forest.

The respective performance of each model is presented below:

1. <u>Bagging Model (Random Forest)</u>:

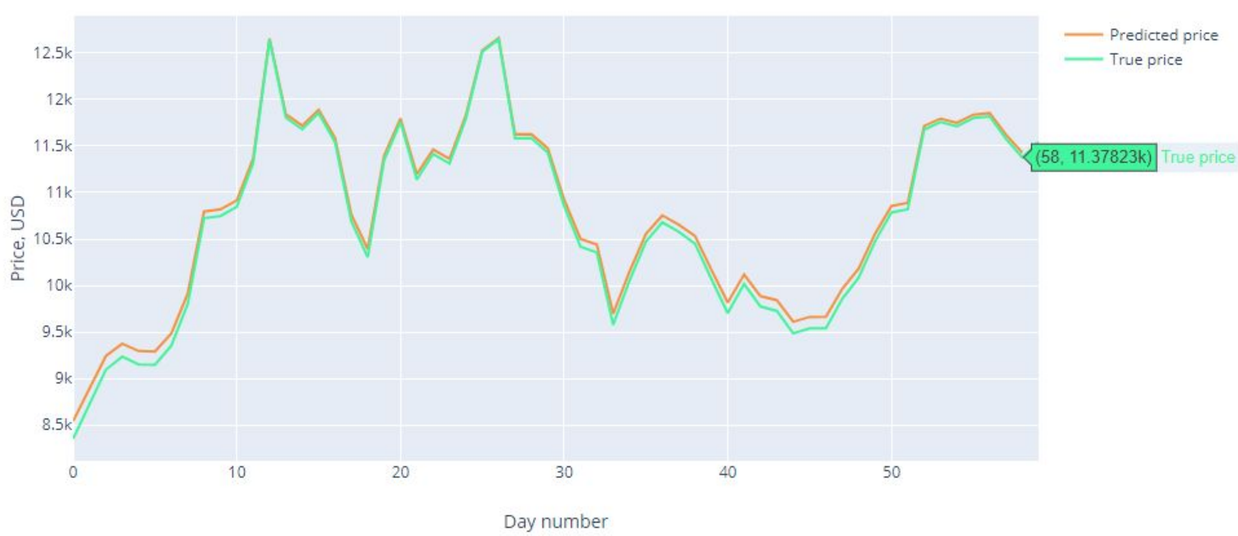


2. <u>Boosting Model (XGBoost)</u>:

Comparison of true prices (on the test dataset) with prices XGBoost model predicted


3. <u>Deep Learning Model (RNN with LSTM)</u>:

Comparison of true prices (on the test dataset) with prices Deep Learning model predicted



**Performance Metric for all models:**

| | Absolute Percentage Error | Root Mean Square Error | R^2 Score (coefficient of determination) | Explained Variance |
|---|---|---|---|---|
| **Random Forest** | 93.690 | 802.672 | 0.325 | 0.513 |
| **XG Boost** | 98.095 | 321.697 | 0.892 | 0.893 |
| **RNN with LSTM** | 99.227 | 87.150 | 0.992 | 0.998 |

# 4. Conclusion

It can now be easily concluded that deep learning algorithms perform significantly better than other machine learning algorithm as far as tasks like forecasting cryptocurrency's price are concerned. The deep learning model proposed here performs better than most of the prevalent models like those in [4] achieving an accuracy of over 99.22%. However, there are several other methodologies and improvements that could be incorporated in this project like ensembling with boosting algorithm. Thus, there is still scope for further improvement and enhancement.

# 5. References

[1] Dataset: https://www.kaggle.com/mczielinski/bitcoin-historical-data/data
[2] About XGBoost & its hyperparameter tuning:
https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/
[3] Wikipedia.com, Recurrent Neural Networks,
https://www.wikizero.pro/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvUmVjdXJyZW50X25ldXJhbF9uZXR3b3Jr
[4] IEEE Paper: https://ieeexplore.ieee.org/document/8374483/authors#authors
[5] When to sell Bitcoin:
https://www.businessinsider.in/Bitcoins-value-is-skyrocketing-heres-how-to-figure-out-if-you-should-sell/articleshow/62060948.cms
[6] LSTM: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[7] Shah, Devavrat, and Kang Zhang. "Bayesian regression and Bitcoin." arXiv preprint arXiv:1410.1231 (2014).
[8] Bitcoin Price Prediction: http://ceur-ws.org/Vol-2280/paper-06.pdf
[9] S. Siami-Namini and A. S. Namin, "Forecasting Economics and Financial Time Series: ARIMA vs. LSTM," https://arxiv.org/abs/1803.06386v1 vol. abs/1803.06386, 2018.
[10] https://arxiv.org/pdf/1907.00558.pdf

Submitted by:
Manav Kaushik
Email | Website | LinkedIn
BITS Pilani- Pilani Campus