Bitcoin Price Prediction using Machine Learning

Purva Kulkarni University of Florida

***Abstract*—In today's world, Bitcoin is amongst the most popular digital cryptocurrencies which is decentralized in nature. It uses Blockchain Technology, which is a distributed and open ledger recording transactions, first implemented in 2009 by Satoshi Nakamoto. Bitcoin currently has the largest capitalization of market. Even though it experienced popular awareness in the past decade, a great amount of volatility was experienced in the Bitcoin value. The durations of elevated volatility make prediction of Cryptocurrency Price a significant, alluring and attractive topic of study which draws investor's interest and massive media attention. This paper predicts the Bitcoin price in USD. The primary objective is to achieve price prediction accuracy using Recurrent Neural Network(RNN) and (LSTM) Long Short Term Memory network. As a comparison to the deep learning model, the traditionally popular Autoregressive Integrated Moving Average (ARIMA) model is studied and analyzed for time series forecasting. ARIMA model is outperformed by the non-linear deep learning model and Long Short Term Memory achieves the highest accuracy. As a part of Time series narrative, exploration of wavelets has been done but not implemented for the purpose of prediction. The model has been implemented on a GPU and a CPU and it has been observed that GPU outperforms CPU training time by almost 68%.**

***Index Terms*— Blockchain, Bitcoin, crypto currency, Recurrent Neural Network, Time series prediction, Long Short Term Memory**

I. Introduction

BITCOIN is a worldwide leading cryptocurrency used for digital payments and purpose of investment. It operates on peer to peer, distributed and trustless model because of which it presents a different paradigm as compared to the traditional financial markets. The creation and transaction of money and it's management is dealt by the peers of the network. There is no centralized authority to control bitcoin. Blockchain technology uses miners and cryptographic proof to verify the Bitcoin transactions which are posted in the form of blocks. This verification does not need any intermediary for the transactions from sender to receiver[9]. Bitcoin currently has market capitalization of approximately 9 billion dollars. In the past decade, Bitcoin transactions have gone over 300,000 per day.

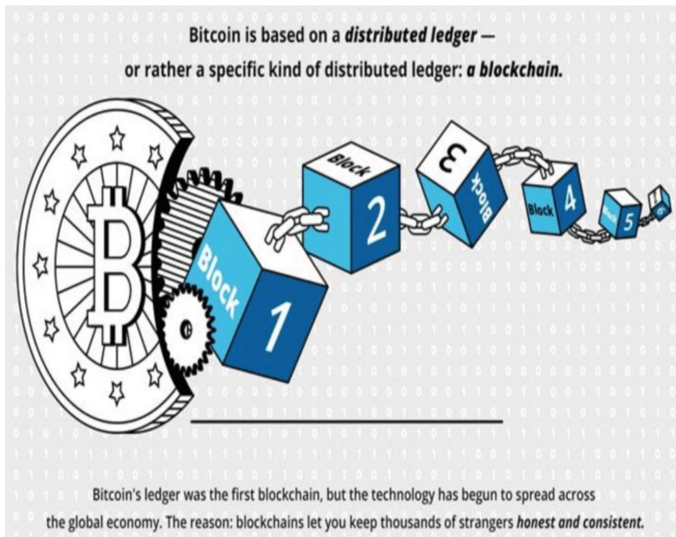


Fig. 1. Blockchain Technology.

Bitcoin poses an interesting problem of Time series prediction when it's market is still in the transitory stage. The Time Series forecasting is a significant prediction task. The research in the field of analysis of Time series and forecasting future values is focused since a very long time. These methods are helpful when seasonal effects are present in the historical data. As Bitcoin lacks this seasonality and due to it's high volatile nature, this method is ineffective. Because of the complexity of the given task, several deep learning techniques make great technological solutions based on the performance. Even though significant amount of research has been conducted in various machine learning techniques, there is a lack in the field related to Bitcoin price prediction. The Long Short Term Memory (LSTM) and (RNN) ie the Recurrent Neural Network are preferred over the traditionally used multilayer perceptron because of the temporal behavior.

The Recurrent Neural Network was developed by Elman and it's structure is similar to traditional multilayer perceptron.

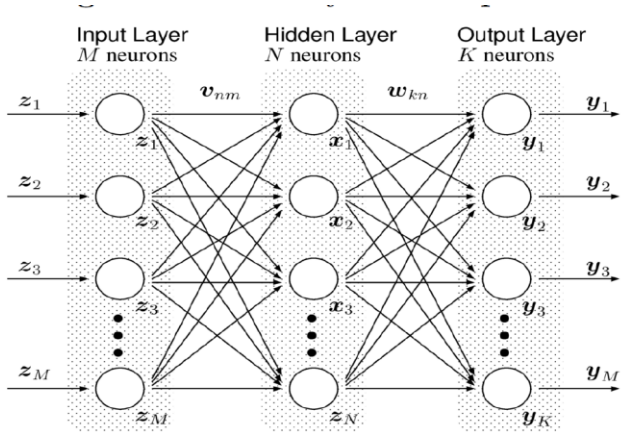


Fig. 1.1. MultiLayer Perceptron

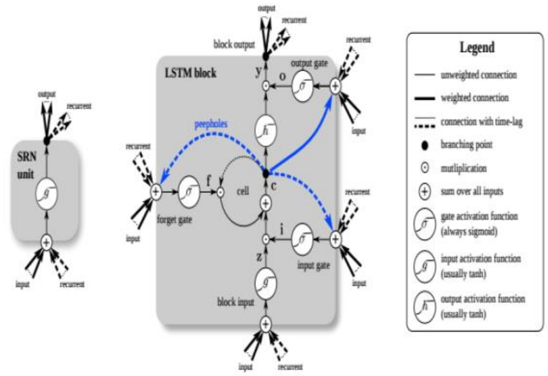


Fig. 3. Long Short Term Memory.

The Context layer facilitates bidirectional flow of signals in an iterative manner. The state is overwritten at every timestep in this context. Long Short Term Memory preserves the forwarded and the back-propagated weights through the layers. The network continues to learn over several timesteps. The network can learn long term dependencies. It has remember and forget gates which permits the network in deciding the information which can be blocked or allowed to be passed depending on the strength and significance. The weak signals are blocked preventing the vanishing gradient.

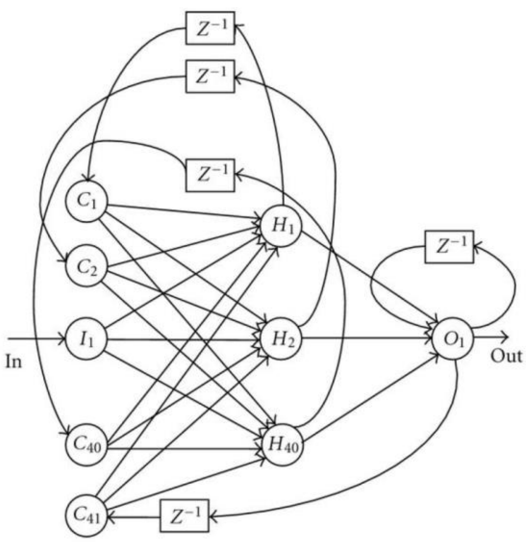


Fig. 2. Recurrent Neural Network

II. Related Work

There is a lack in research on prediction of Bitcoin price based on machine learning techniques. [5]An inactive source model was implemented by Shah et. Al for prediction of Bitcoin price. In 50 days, the model had received Sharpe ratio of 4.1 and a return of 89 percent. Investigation of the determinants and implementation of sentiment analysis was carried out by Geourgoula et al. with the help of Support Vector Machines. He found a positive correlation between frequency of views on Wikipedia and hash rate of network. [10] The relationship between tweets, coin price and bitcoin views was investigated by Matta et al. A weak to medium correlation was found out between views and positive tweets and the Bitcoin price. This was a used as a proof to claim that they can be used as predictors. However, 60 days was the sample size which was a major disadvantage. The sentiment was a variable which can spread misinformation on social media platforms like Twitter or Reddit. The market can suffer a great threat of manipulation because of limited liquidity exchanges. Therefore, sentiment was ruled out and was no longer considered.

An accuracy of 55% was reported with Artificial Neural Network by Greaves et al with Blockchain bitcoin[8]. Random forests, Binomial GLM and Support Vector Machines were implemented using the Blockchain data with accuracy of 97%. However, these results were not cross validated. Since the development and use of the back propagation algorithm, a lot of research was conducted to use it in solving these problems[7]. One of the major disadvantages of using Multilayer Perceptron is that the global maximum cannot be guaranteed.

Rather et al. incorporated Recurrent Neural Network in a hybrid model with a genetic algorithm in order to optimize the selection of weights[4]. The results were quite successful and it was found that this network was optimal. Long Short Term Memory can choose the data which needs to be remembered and the data which can be forgotten based on significance and the weights of the parameters. But a lot of computation is required by Long Short Term Memory and RNN which is a disadvantage. Further research showed that the processing was faster on GPUs instead of CPUs. Because of CUDA processing by NVIDIA in 2006, parallel computation has increased the computation speed.

III. Methodology

The implementation of this project reflects Agile method a it follows incremental approach of 4 levels of iterative nature in which the tasks are completed.

A. Feature Extraction and Engineering:

Extraction of features is the method of extracting meaningful information from a huge pool of data to make prediction using machine learning models easier. It is one of the primary and the most important step in prediction tasks. Good feature extraction will result in accurate predictions. One should have good domain knowledge. The features which are engineered should be able to represent the things which are being taught to the network. But the features which are chosen should be evaluated properly to make sure that the predictability of the datasets is improved. There are additional technical quantitative indicators which provide a description of previous trends numerically.

Deep Learning models do not demand much feature engineering as the hidden layers learn from the features which are not linear in nature themselves and detect long term dependencies in Long Short Term Memory in sequential data[2]. But it is still important to carry out feature engineering because there is no way to know what the model has learned. Feature Engineering is a subjective process and one can explicitly chose features which are considered important.

B. Feature Evaluation:

Once the features are selected, they are evaluated because the large set of features will reduce the accuracy of the model and increase the training time considerably. There are many feature evaluation techniques based on wrapper based selection and filter based selection. Wrapper based selection methodology carry out heuristic search of the solutions to the classifier whereas Filter based selection is based on the statistical properties of the feature.

One of the wrapper based selection algorithm is the Boruta algorithm which is build around Random Forests Classification. The principle of this algorithm is similar to Random Forests by adding randomness in the model and collecting results from randomized samples for attribute evaluation. This tells the important features out of all the feature which have been extracted. Principal Component Analysis (PCA) was studied and explored for dimensionality reduction.

C. Recurrent Neural Network(RNN):

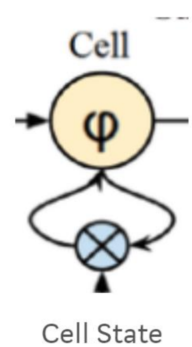
A recurrent neural network is developed for recognition of sequential sets of data such as genomes, handwriting, texts and helps in predicting patterns. Recurrent Neural Networks take the current input and the previously timed input as well. Therefore the current input and the recent past input is fed into the network.
The sequential data is preserved in the hidden layers which manages to traverse several steps forward in a cascading manner which affects the every sample in the future. The step of processing the samples forward is given by:

$$\mathbf{h}_t = \phi\left(W\mathbf{x}_t + U\mathbf{h}_{t-1}\right)$$

D. Long Short Term Memory (LSTM):

Long Short Term Memory is capable of learning the long-term dependencies. They selectively remember patterns for a duration of time. LSTM modifies the information by performing multiplications and additions. There are cell states present through which the information flows. Every memory cell has an architecture internally which guarantees constant error. Memory blocks are known as sets of blocks which are connected recurrently.

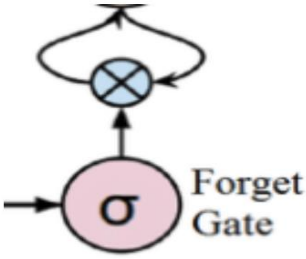
The modification of cell state takes place because of the forget gate in the network which is placed below the call state and it is adjusted with the help of input modulation gate. Input gate is the save vector and the remember vector is the forget gate. If the output of the forget gate is 0 then the information is blocked and if it is 1, the information will be kept in the cell. The focus vector is known as the output gate.



$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t$$

Cell State Equation

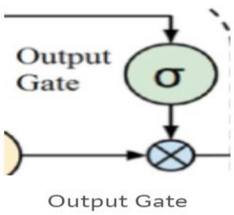
Fig. 4. Cell state Equation.



$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_t\right)$$

Forget Gate Equation

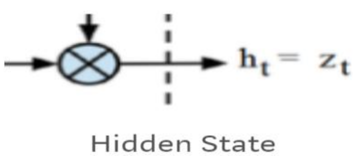
Fig. 5. Forget Gate Equation.



$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\right)$$

Output Gate Equation

Fig. 6. Output Gate Equation



$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$

Hidden State Equation

Fig. 7. Hidden State Equation.

IV. Implementation

After feature scaling, the dataset is further divided into 2 separate sets: training and test sets. The given data is reshaped according to the requirement of the model. The data is scaled by the MinMaxScaler because the data is scale sensitive. Keras framework has been used for LSTM deep learning model.

The number of hidden layers were two with every layer having 256 units. It is densely connected with a single output neuron. The system stops early if the expected result doesn't improve in the 15 epochs. Several combinations of epochs and batch size were tested and finally 100 epochs and a batch size of 256 was decided. Usually, two layers are enough for a such type of task. Testing of the LSTM network with three and four hidden layers was done but it did not enhance the performance of the system.

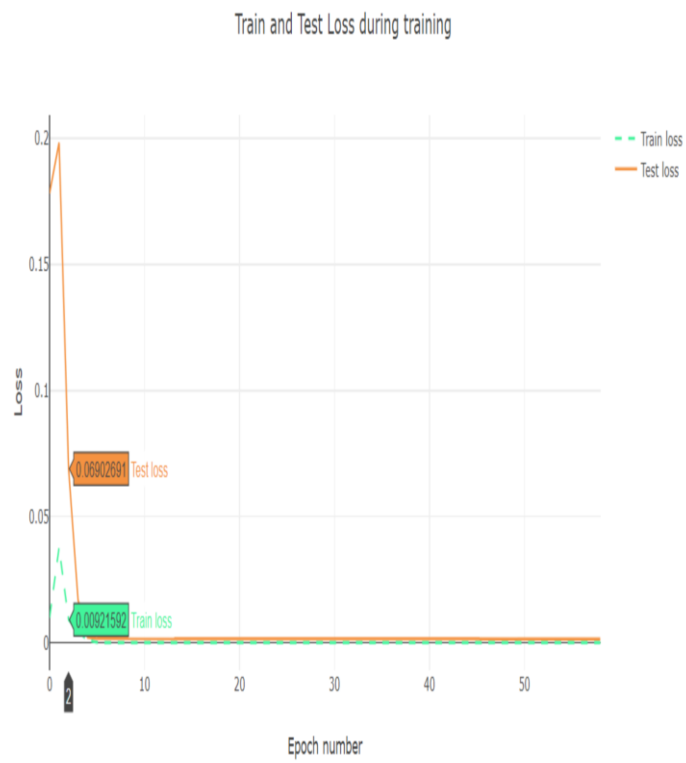


Fig. 8. Train and test loss during training.

A. *Model Comparison:*

A confusion matrix is constructed using true-false and positive and negative values. The accuracy of the system is the number of correctly predicted values. However only accuracy is not enough as the data might be misleading. To ensure the correctness, sensitivity, precision and specificity is also calculated. Evaluation and comparison of regression accuracy is done by Root Mean Square Error.

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{FP + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP + TN}{P + N}$$

$$RMSE = \sqrt{(xi - yi)^2}$$

Fig. 9. Formulae

B. Validation:

The default validation is the holdout validation. The model is trained on 80% of the available dataset and the remaining 20% dataset is used for validation. K-fold cross validation was studied but not implemented because of the heavy training time. Another validation method, called as the sliding window validation performed poorly in Theano version.

C. Autoregressive Integrated Moving Average:

It is popular methodology used for forecasting of time series. Depending on the number of time lags, time series is divided into models. The function auto.arima fits the data into the possibly best order and the other one is determined based PACF and ACF plots.

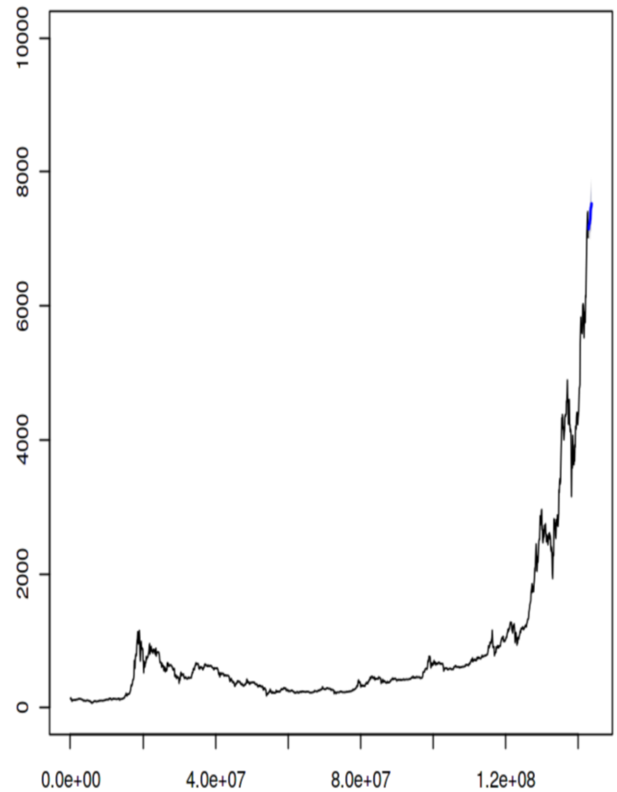


Fig. 9.1. Forecasts from ARIMA

V. Evaluations

The ARIMA model performed poorly with lowest accuracy and RMSE. It was predicted that the price would go up each day and there was a lack of false positives because of the class imbalance. But the ARIMA model performed better on sensitivity, precision and specificity. The LSTM model achieved the highest accuracy and lowest Root Mean Square Error. After validating the data, LSTM got a error rate of 8.07%. The training time of LSTM was considerably low. The sensitivity of the LSTM model was 37% whereas the specificity was 61.30%. The Precision was 35.50% and RMSE was 6.87% On the other hand, the ARIMA model showed a precision of 1 and it's specificity is also 1. The sensitivity is 14.7% of ARIMA model and the Root Mean Square Error is 53.74%.

The CPU which was used in the performance evaluation was Intel core i7 which is quite high specification processor and the GPU used is NVIDIA GeForce. The GPU outperformed CPU significantly. The GPU was almost 68% percent faster than the computation on CPU. The LSTM model showed enhanced performance of about 70% on GPU.

Sr no	Attribute	ARIMA	LSTM
1	Sensitivity	14.7%	37%
2	Specificity	1	61.30%
3	Precision	1	35.50%
4	RMSE	53.74%	6.87%

Fig. 10. Evaluation of LSTM and ARIMA models

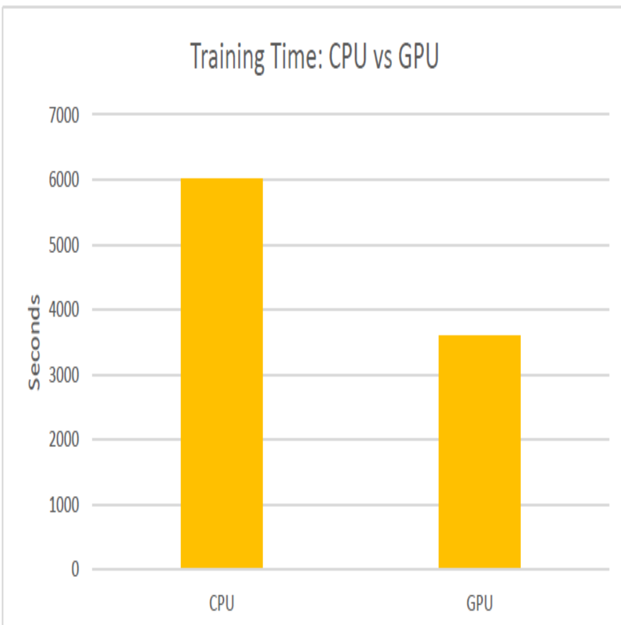


Fig. 11. Training Time CPU vs GPU

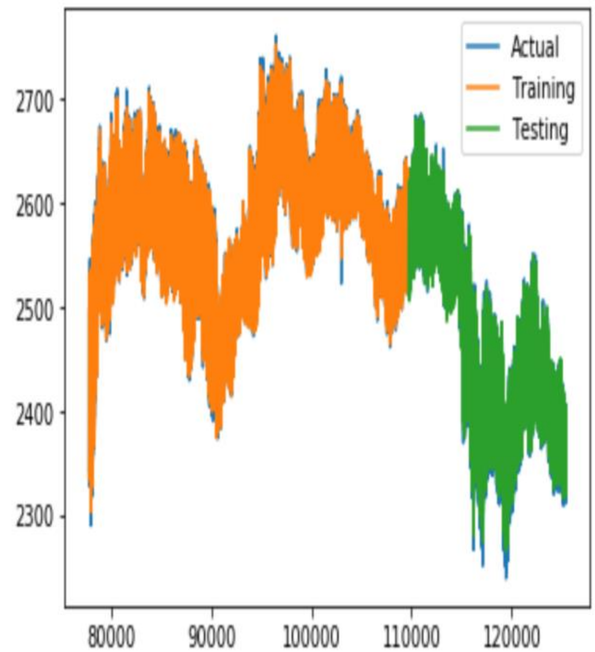


Fig. 12. Prediction of Bitcoin Price

VI. Conclusion

Deep learning models are much more effective learners on the training dataset. The LSTM outperformed ARIMA model and the traditional RNN model due to it's ability to store long term dependencies. However due to the high degree of variance in the Bitcoin data, it is difficult to obtain good and suitable validation. The parallelization for machine learning techniques using GPU give us better results and a performance enhancement of 70.7%.

VII. Future Work

Several other models can be investigated and the best of them can be used to predict the price of Bitcoin. More historical data can be collected to train the model in a better way.

VIII. Acknowledgements

I would like to take this opportunity to thank University of Florida and thus express my deepest gratitude to Professor Dapeng Wu for accompanying and guiding me throughout the project. I got to learn tremendous Pattern Recognition concepts in such short span of time. I would also like to thank my parents for their support and motivation.

IX. REFERENCES

[1] M. Briere, K. Oosterlinck, and A. Szafarz, “Virtual currency, tangible ` return: Portfolio diversification with bitcoins,” Tangible Return: Portfolio

[2] Y. Bengio, “Learning deep architectures for ai,” Foundations and trendsR in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009.

[3] B. Scott, “Bitcoin academic paper database,” suitpossum blog, 2016.

[4] D. K. Wind, “Concepts in predictive machine learning,” in Maters Thesis, 2014

[5] A. Greaves and B. Au, “Using the bitcoin transaction graph to predict the price of bitcoin,” 2015.

[6] H. White, “Economic prediction using neural networks: The case of ibm daily stock returns," in Neural Networks, 1988., IEEE International Conference on. IEEE, 1988, pp. 451, 458.

[7] A. Greaves and B. Au, “Using the bitcoin transaction graph to predict the price of bitcoin,” 2015.

[8] I. Kaastra and M. Boyd, “Designing a neural network for forecasting nancial and economic time series," Neurocomputing, vol. 10, no. 3, pp. 215{236, 1996.

[9] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[10] M. Matta, I. Lunesu, and M. Marchesi, “Bitcoin spread prediction using social and web search media,” Proceedings of DeCAT, 2015. “The predictor impact of web search media on bitcoin trading volumes.”