e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

PREDICTING THE PRICE OF BITCOIN USING MACHINE LEARNING

Yannaru Paradesi*1, Suneetha A*2

*1,2Student, Department Of MCA, Sree Vidyanikethan Institute Of Management,
Tirupati, Andhra Pradesh, India.

DOI : https://www.doi.org/10.56726/IRJMETS30998

ABSTRACT

The goal of this paper is to ascertain with what accuracy the direction of Bitcoin price in USD can be predicted. The price data is sourced from the Bitcoin Price Index. The task is achieved with varying degrees of success through the implementation of a Bayesian optimized recurrent neural network (RNN) and a Long Short Term Memory (LSTM) network. The LSTM achieves the highest classification accuracy. The popular ARIMA model for time series forecasting is implemented as a comparison to the deep learning models. As expected, the non-linear deep learning methods outperform the ARIMA forecast which performs poorly. Finally, both deep learning models are benchmarked on both a GPU and a CPU with the training time on the GPU outperforming the CPU implementation by 67.7%.

Keywords: Bitcoin, Deep Learning, GPU, Recurrent Neural Network, Long And Short Term Memory, ARIMA.

I. INTRODUCTION

Bitcoin is the worlds' most valuable cryptocurrency and is traded on over 40 exchanges worldwide accepting over 30 different currencies. It has a current market capitalization of 9 billion USD according to https://www.blockchain.info/ and sees over 250,000 transactions taking place per day. As a currency, Bitcoin offers a novel opportunity for price prediction due its relatively young age and resulting volatility, which is far greater than that of fiat currencies. It is also unique in relation to traditional fiat currencies in terms of its open nature; no complete data exists regarding cash transactions or money in circulation for fiat currencies. Prediction of mature financial markets such as the stock market has been researched at length. Bitcoin presents an interesting parallel to this as it is a time series prediction problem in a market still in its transient stage. Traditional time series prediction methods such as Holt-Winters's exponential smoothing models rely on linear assumptions and require data that can be broken down into trend, seasonal and noise to be effective. This type of methodology is more suitable for a task such as forecasting sales where seasonal effects are present. Due to the lack of seasonality in the Bitcoin market and its high volatility, these methods are not very effective for this task. Given the complexity of the task, deep learning makes for an interesting technological solution based on its performance in similar areas. The recurrent neural network (RNN) and the long short-term memory (LSTM) are favored over the traditional multilayer perceptron (MLP) due to the temporal nature of Bitcoin data.The aim of this paper is to investigate with what accuracy the price of Bitcoin can be predicted using machine learning and compare parallelization methods executed on multi-core and GPU environments. This paper contributes in the following manner: of approximately 653 papers published on Bitcoin, only 7 (at the time of writing) are related to machine learning for prediction. To facilitate a comparison to more traditional approaches in financial forecasting, an ARIMA time series model is also developed for performance comparison purposes with the neural network models.

II. LITERATURE REVIEW

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008

A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and

outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

[2] M. Briere, K. Oosterlinck and A. Szafarz "Virtual currency, tangible ` return: Portfolio diversification with bitcoins," Tangible Return: Portfolio Diversification with Bitcoins (September 12, 2013), 2013.

Bitcoin is a major virtual currency. Using weekly data over the 2010-2013 period, we analyses a Bitcoin investment from the standpoint of a U.S. investor with a diversified portfolio including both traditional assets (worldwide stocks, bonds, hard currencies) and alternative investments (commodities, hedge funds, real estate). Over the period under consideration, Bitcoin investment had highly distinctive features, including exceptionally high average return and volatility. Its correlation with other assets was remarkably low. Spanning tests confirm that Bitcoin investment offers significant diversification benefits. We show that the inclusion of even a small proportion of Bitcoins may dramatically improve the risk-return trade-off of well-diversified portfolios. Results should however be taken with caution as the data may reflect early-stage behaviour which may not last in the medium or long run.

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

Artificial neural networks are universal and highly flexible function approximators first used in the fields of cognitive science and engineering. In recent years, neural network applications in finance for such tasks as pattern recognition, classification, and time series forecasting have dramatically increased. However, the large number of parameters that must be selected to develop a neural network forecasting model have meant that the design process still involves much trial and error. The objective of this paper is to provide a practical introductory guide in the design of a neural network for forecasting economic time series data. An eight-step procedure to design a neural network forecasting model is explained including a discussion of tradeoffs in parameter selection, some common pitfalls, and points of disagreement among practitioners.

[4] 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.

A report is presented of some results of an ongoing project using neural-network modeling and learning techniques to search for and decode nonlinear regularities in asset price movements. The author focuses on the case of IBM common stock daily returns. Having to deal with the salient features of economic data highlights the role to be played by statistical inference and requires modifications to standard learning techniques which may prove useful in other contexts.

III. EXISTING SYSTEM AND PROPOSED SYSTEM

To fully understand just how powerful live forecasting is, we need to look at it from the very beginning which is traditional forecasting. Traditional forecasting basically uses historical observations to estimate future business metrics like inventory requirements, budgets, revenue and asset performance. Traditional forecasting practices fail because the past does not necessarily represent the future. Machine Learning have different types of algorithms and each has its own structure of working procedure which contains different intuitions. In these intuitions, models are working on different procedures in different ways and also delivers high and less accuracies. This creates a lot of constraints regarding cases, fatalities and recoveries available. The major challenge is to create a model for them so that no one have less accuracy. In this existing system we used KNN and Logistic regression which results less accuracy. Such a models won't be able to match with our requirements and it also consumes more time.

DISADVANTAGES:

- Low efficiency.
- Time consuming.
- High complexities.
- Resources consuming

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

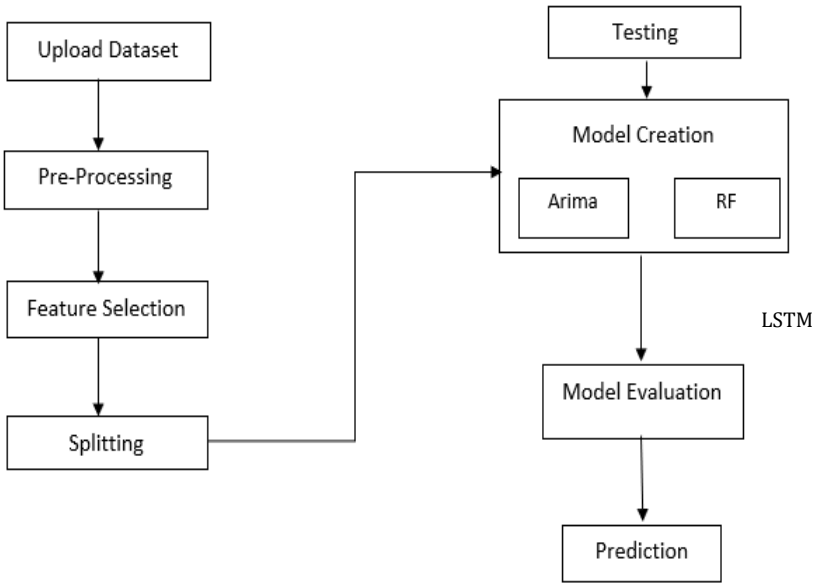

Fig-1: Flow of the project

We propose this application that can be considered a useful system since it helps to reduce the limitations obtained from KNN and Logistic Regression. By providing support through the forecasting analysis, it can be able to generate best results for attributes without any overlap. Models involved in this application are ARIMA and LSTM models.

ADVANTAGES:

- High efficiency.
- Time Saving.
- Inexpensive.
- Low complexities

IV. ARCHITECTURE AND MODULES FOR OVERALL PROCESS

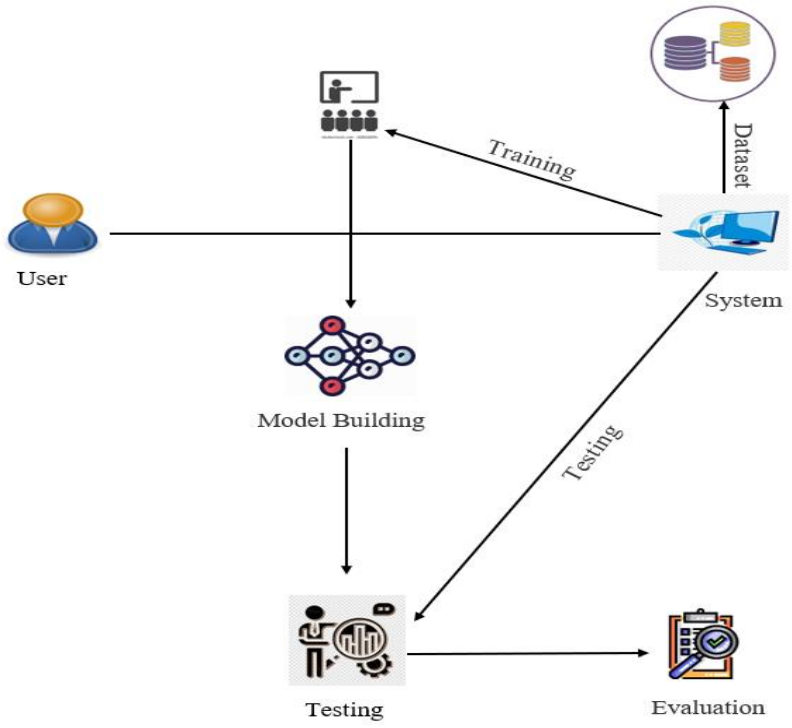

Fig-2: Architecture of overall process

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

System

User

1.System:

1.1 Receive Dataset:

Receive Dataset from the user

1.2 Pre-processing:

Perform pre-processing on data set

1.3Training:

Use the pre-processed training dataset to train our model using LGBM algorithms.

1.4 Generate Results:

View Results generated

2.User:

2.1 Upload

The user needs to upload the data.

2.2 View-Data

Later on, user can upload data

2.2 View Pre-processing:

Later on, user can view the pre-processing of data.

2.3View training:

Later on, user can view the training of data.

V. ALGORITHMS AND THEIR APPLICAIONS

ARIMA:

In statistics and econometrics, and in particular in time series analysis, an autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity in the sense of mean (but not variance/autocovariance), where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity of the mean function (i.e., the trend). When the seasonality shows in a time series, the seasonal-differencingcould be applied to eliminate the seasonal component.The AR part of ARIMA indicates that the evolving variable of interest is regressed on its own lagged (i.e., prior) values. The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past.The I (for "integrated") indicates that the data values have been replaced with the difference between their values and the previous values (and this differencing process may have been performed more than once). The purpose of each of these features is to make the model fit the data as well as possible.

APPLICATIONS:

In business and finance, the ARIMA model can be used to forecast future quantities (or even prices) based on historical data. Therefore, for the model to be reliable, the data must be reliable and must show a relatively long time span over which it's been collected. Some of the applications of the ARIMA model in business are listed below:

- Forecasting the quantity of a good needed for the next time period based on historical data.
- Forecasting sales and interpreting seasonal changes in sales
- Estimating the impact of marketing events, new product launches, and so on.

LSTM:

Feed-Forward Neural Networks:

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

- A feed-forward neural network allows information to flow only in the forward direction, from the input nodes, through the hidden layers, and to the output nodes. There are no cycles or loops in the network.
- Below is how a simplified presentation of a feed-forward neural network looks like:

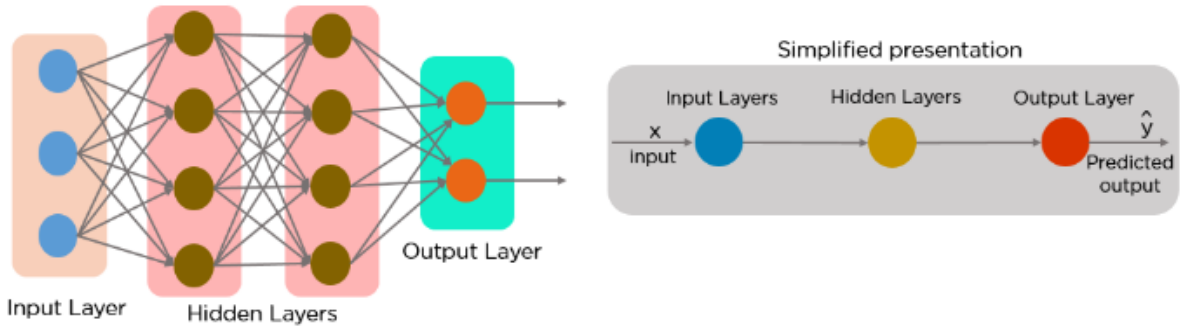

Fig-3: Feed-forward neural network

Why Recurrent Neural Networks are better?

- Recurrent neural networks were created because there were a few issues in the feed-forward neural network:
- Cannot handle sequential data
- Considers only the current input
- Cannot memorize previous inputs
- The solution to these issues is the Recurrent Neural Network (RNN). An RNN can handle sequential data, accepting the current input data, and previously received inputs. RNNs can memorize previous inputs due to their internal memory.

Applications of RNN:

- NLP
- Time series
- Language Translation.

LSTM:

- Why Recurrent Neural Networks?
- Recurrent neural networks were created because there were a few issues in the feed-forward neural network:
- Cannot handle sequential data
- Considers only the current input
- Cannot memorize previous inputs
- The solution to these issues is the Recurrent Neural Network (RNN). An RNN can handle sequential data, accepting the current input data, and previously received inputs. RNNs can memorize previous inputs due to their internal memory.
- It is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data, apart from single data points such as images.
- The central role of an LSTM model is held by a memory cell known as a 'cell state' that maintains its state over time. The cell state is the horizontal line that runs through the top of the below diagram. It can be visualized as a conveyor belt through which information just flows, unchanged.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

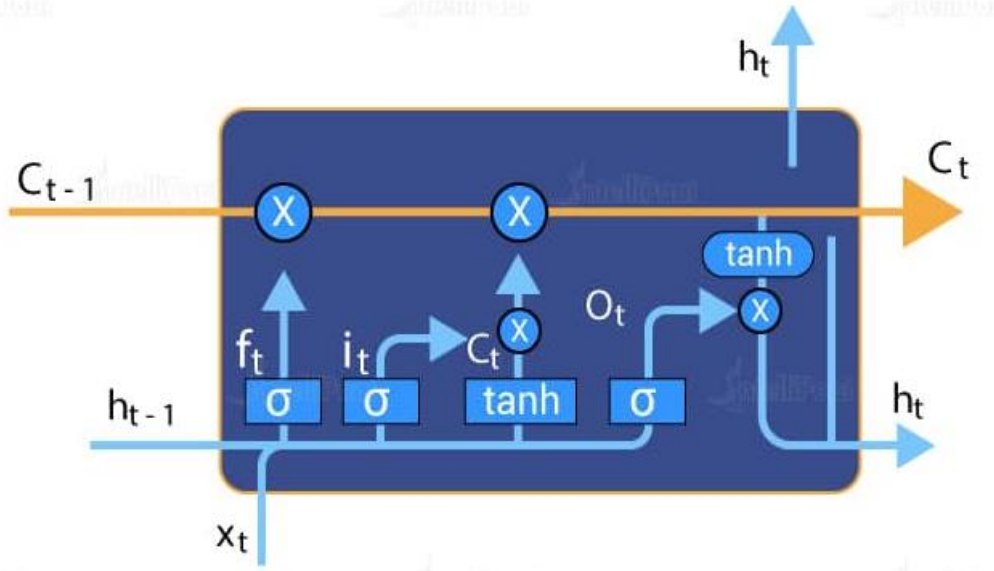

Fig-4: LSTM sigmoid neural net layer

- Information can be added to or removed from the cell state in LSTM and is regulated by gates. These gates optionally let the information flow in and out of the cell. It contains a pointwise multiplication operation and a sigmoid neural net layer that assist the mechanism.
- The remember vector is usually called the **forgetgate**. The output of the forget gate tells the cell state which information to forget by multiplying 0 to a position in the matrix. If the output of the forget gate is 1, the information is kept in the cell state. From equation, sigmoid function is applied to the weighted input/observation and previous hidden state.

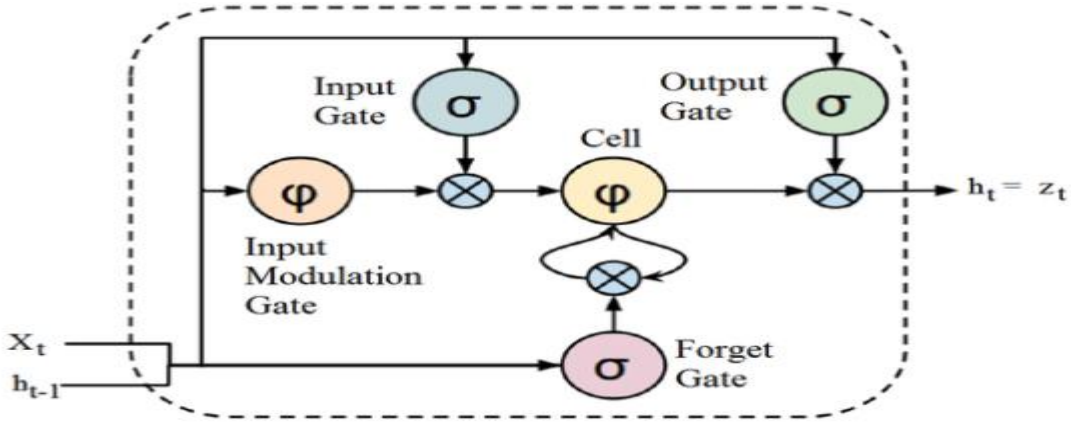

Fig-5: LSTM cell

- The save vector is usually called the **inputgate**. These gates determine which information should enter the cell state / long-term memory. The important parts are the activation functions for each gates.
- The input gate is a sigmoid function and have a range of [0,1]. Because the equation of the cell state is a summation between the previous cell state, sigmoid function alone will only add memory and not be able to remove/forget memory.
- If you can only add a float number between [0,1], that number will never be zero / turned-off / forget. This is why the input modulation gate has an tanh activation function. Tanh has a range of [-1, 1] and allows the cell state to forget memory.

LSTM Applications:

LSTM networks find useful applications in the following areas:

- Language modeling
- Machine translation

- Handwriting recognition
- Image captioning
- Image generation using attention models.

VI. FUTURE SCOPE

This application can be extended to include the ability to predict the future prices. we plan to explore the prediction methodology using the updated dataset and use the most accurate and appropriate methods for forecasting. Real-time live forecasting will be one of the primary focus in our future work.

VII. CONCLUSION

In this application, we have successfully created a model to generate future outcomes for crypto currency called Bitcoin. This is developed in a user-friendly environment using Python programming.

VIII. REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] M. Briere, K. Oosterlinck, and A. Szafarz, "Virtual currency, tangible `return: Portfolio diversification with bitcoins," Tangible Return: Portfolio Diversification with Bitcoins (September 12, 2013), 2013.

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

[4] 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.

[5] C. Chatfield and M. Yar, "Holt-winters forecasting: some practical issues," The Statistician, pp. 129–140, 1988.

[6] B. Scott, "Bitcoin academic paper database," suitpossum blog, 2016.

[7] M.D. Rechenthin, "Machine-learning classification techniques for the analysis and prediction of high-frequency stock direction," 2014.