Bitcoin Price Prediction

Karnati Vinay **Kanna Lakshmi Dathatreya** **Rentala Naga Sai Ganesh** **Aryan Patel**

Abstract: Cryptocurrencies are becoming a major income source because of their high availability and plenty of easy investment platforms. It is a suburbanized system for confirming that the parties to a dealing have the money they claim to possess, eliminating the requirement for ancient intermediaries, like banks, once funds are being transferred between 2 entities. There are a large number of virtual currencies such as bitcoin, ripple, Ethereum, Ethereum classic, lite coin, etc. In this study, we especially focused on a in demand cryptocurrency, i.e., bitcoin. The forecasting is done using different time series analysis techniques like moving average, ARIMA and machine learning algorithms including SVM, and various other models such as LSTM and Prophet.

Keywords: bitcoin, cryptocurrency, ARIMA, SVM, LSTM, and Prophet.

1. Introduction

A cryptocurrency, crypto-currency, or crypto could be a digital currency designed to figure as a medium of exchange through a computer network that's not dependent on any central authority, like a bank or government, to endorse or maintain it. The viability of every cryptocurrency's coins is provided by a blockchain. A blockchain could be a endlessly growing list of records, known as blocks, that are connected and secured by the cryptography. usually each and every block contains its own hash pointer this creates a link to a previous block, a timestamp and dealing information(transaction data). By design, blockchains are inherently proof against modification of the information .Each and every dealing is stored, and multiple copies of same Ledger is hold on. Bitcoin is the one in every of the various cryptocurrencies are offered these days.

Since cryptocurrency, particularly Bitcoin, dealing volumes became high and memorable for the last 3 years, the correct prognostication of future values of those are becoming a lot of vital and attention-grabbing space for each researcher and investors. As Bitcoin has been a financial plus and is listed through several cryptocurrency exchanges sort of a securities market, several researchers have investigated varied factors that have an effect on the Bitcoin worth and therefore the patterns behind its fluctuations victimization numerous analytical and experimental ways. during this Paper we'll study numerous bitcoin postulation models like ARIMA, SUPPORT VECTOR MACHINE(SVM), Long Short-Term Memory (LSTM) and Prophet.

This paper is organized as follows. section II briefly explains the literature survey. The dataset, information preprocessing, and information analytics steps square measure represented in Section III. Experiments and results square measure given in section IV, whereas the conclusions square measure given in section V and later References

2. Literature Survey

Temesgen Awoke, Minakhi Rout, LipikaMohanty, and Suresh Chandra Satapathyhas Proposed a research paper on bitcoin price prediction and analysis using deep learning models such as long short-term memory (LSTM) and gated recurrent unit (GRU) to handle the price volatility of bitcoin and to obtain high accuracy. Their study involves comparing these two-time series deep learning techniques and proved the efficacy in forecasting the price of bitcoin. The study reveals that the GRU model is the better mechanism for time series cryptocurrency price prediction and takes lower compilation time.

IsılYenido˘gan, AykutCayır,OzanKozan, TugceDag, Cigdem Arslan has Proposed a research paper on bitcoinforecasting using statistical analysis models such as ARIMA and PROPHET. To find the most accurate forecast model, the performance metrics of PROPHET and ARIMA methods are compared on the same dataset. The model selection for both ARIMA and PROPHET is done by using threefold splitting technique considering the time series characteristics of the dataset.

Finally, two different models are created and compared in terms of performance metrics and they have concluded that PROPHET model outperforms ARIMA model 0.94 to 0.68 in R2 values

Suhwan Ji, Jongmin Kim and HyeonseungIm has Proposed a research paper based on a comparative study of bitcoin price prediction using Deep Learning. In the paper, they have studied and compared various state-of-the-art Deep Learning methods such as a deep neural network (DNN), a long short-term memory (LSTM)model, a convolutional neural network, a deep residual network, and their combinations for Bitcoin price prediction and experimental results showed that although LSTM-based prediction models slightly outperformed the other prediction models for Bitcoin price prediction (regression), DNN-based models performed the best for price ups and downs prediction (classification). They have concluded that classification models were more effective than regression models for algorithmic trading

Grace. LK. Joshila, Asha. P, D. Usha Nandini, G. Kalaiarasi has Proposed a research paper price prediction of bitcoin using the machine learning algorithm such as SVM (support vector machine). By using keras function it is easy to calculate or manipulating the number functions, even it will take less time to train data when compared to other function and another advantage is time complexity is low when compared to Convolutional neural network and random forest.so when they checked for accuracy, they have concluded that support vector machine (SVM) is much more accurate than any other algorithms.

3. Data Preprocessing and Data Analysis

Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or any other algorithms. The major steps involved in the data preprocessing are data cleaning, data integration, data reduction and data transformation.

3.1 DATA CLEANING

Data cleaning is the process to remove incorrect data, incomplete data and inaccurate data from the datasets, and it also replaces the missing values. There are some techniques in data cleaning like we can remove the missing values or we can replace the missing values with the methods such as mean, median and mode. This process is known as imputation

3.1.1 ARIMA data set

We are using imputation by mean for cleaning the data.

ds	yL	yH	yO	y	Volume	Marketcap
6/13/2015 23:59	232.652	229.210	229.920	232.402	13305300	3316504361
6/14/2015 23:59	234.858	232.004	232.442	233.543	12165900	3333657006
6/15/2015 23:59	237.836	233.422	233.422	236.823	19912100	3381329191
6/16/2015 23:59	251.742	236.122	236.765	250.895	41612000	3582987588
6/17/2015 23:59	256.853	246.476	250.823	249.284	43858400	3560953387
6/18/2015 23:59	252.108	244.127	249.428	249.007	30980200	3557986321

```
> ARIMA$yL = ifelse(is.na(ARIMA$yL),ave(ARIMA$yL, FUN = function(x) mean(x, na.rm = 'TRUE')),ARIMA$yL)
> head(ARIMA$yL)
[1] 232.652 234.858 237.836 251.742 256.853 252.108
> ARIMA$yH = ifelse(is.na(ARIMA$yH),ave(ARIMA$yH, FUN = function(x) mean(x, na.rm = 'TRUE')),ARIMA$yH)
> head(ARIMA$yH)
[1] 229.210 232.004 233.422 236.122 246.476 244.127
> ARIMA$yO = ifelse(is.na(ARIMA$yO),ave(ARIMA$yO, FUN = function(x) mean(x, na.rm = 'TRUE')),ARIMA$yO)
> head(ARIMA$yO)
[1] 229.920 232.442 233.422 236.765 250.823 249.428
> ARIMA$y = ifelse(is.na(ARIMA$y),ave(ARIMA$y, FUN = function(x) mean(x, na.rm = 'TRUE')),ARIMA$y)
> head(ARIMA$y)
[1] 232.402 233.543 236.823 250.895 249.284 249.007
> ARIMA$Volume = ifelse(is.na(ARIMA$Volume),ave(ARIMA$yO, FUN = function(x) mean(x, na.rm = 'TRUE')),ARIMA$Volume)
> head(ARIMA$Volume)
[1] 13305300 12165900 19912100 41612000 43858400 30980200
> ARIMA$Marketcap = ifelse(is.na(ARIMA$Marketcap),ave(ARIMA$yO, FUN = function(x) mean(x, na.rm = 'TRUE')),ARIMA$Marketcap)
> head(ARIMA$Marketcap)
[1] 3316504361 3333657006 3381329191 3582987588 3560953387 3557986321
```

3.1.2 SVM Dataset

	ds	yL	yH	yO	yC
1	5/27/2021 23:59	40379.62	37247.90	39316.89	38436.97
2	5/28/2021 23:59	38856.97	34779.04	38507.08	35697.61
3	5/29/2021 23:59	37234.50	33693.93	35684.16	34616.07
4	5/30/2021 23:59	36400.67	33520.74	34607.41	35678.13
5	5/31/2021 23:59	37468.25	34241.94	35658.59	37332.85
6	6/1/2021 23:59	37896.74	35787.09	37293.79	36684.92

```
> view(head(svm))
> svm$yL = ifelse(is.na(svm$yL),ave(svm$yL, FUN = function(x) mean(x, na.rm = 'TRUE')),svm$yL)
> head(svm$yL)
[1] 40379.62 38856.97 37234.50 36400.67 37468.25 37896.74
> svm$yH = ifelse(is.na(svm$yH),ave(svm$yH, FUN = function(x) mean(x, na.rm = 'TRUE')),svm$yH)
> head(svm$yH)
[1] 37247.90 34779.04 33693.93 33520.74 34241.94 35787.09
> svm$yO = ifelse(is.na(svm$yO),ave(svm$yO, FUN = function(x) mean(x, na.rm = 'TRUE')),svm$yO)
> head(svm$yO)
[1] 39316.89 38507.08 35684.16 34607.41 35658.59 37293.79
> svm$yC = ifelse(is.na(svm$yC),ave(svm$yC, FUN = function(x) mean(x, na.rm = 'TRUE')),svm$yC)
> head(svm$yC)
[1] 38436.97 35697.61 34616.07 35678.13 37332.85 36684.92
>
```

3.1.3 Prophet Dataset

	ds	y
1	2015-06-13	232.402
2	2015-06-14	233.543
3	2015-06-15	236.823
4	2015-06-16	250.895
5	2015-06-17	249.284
6	2015-06-18	249.007

```
> bitcoin$ds = ifelse(is.na(bitcoin$ds),ave(bitcoin$ds, FUN = function(x) mean(x, na.rm = 'TRUE')),bitcoin$ds)
> head(bitcoin$ds)
[1] "2015-06-13" "2015-06-14" "2015-06-15" "2015-06-16" "2015-06-17" "2015-06-18"
> bitcoin$y = ifelse(is.na(bitcoin$y),ave(bitcoin$y, FUN = function(x) mean(x, na.rm = 'TRUE')),bitcoin$y)
> head(bitcoin$y)
[1] 232.402 233.543 236.823 250.895 249.284 249.007
>
```

3.2 DATA INTEGRATION

The process of combining multiple sources of data into a single dataset. The Data integration process is one of the main components in data management. There are some problems to be considered during data integration. They are **Schema integration**: Integrates metadata (a set of data that describes other data) from different sources. **Entity identification problem:** Identifying entities from multiple databases. For example, the system or the use should know student _id of one database and student name of another database belongs to the same entity. **Detecting and resolving data value concepts**: The data taken from different databases while merging may differ. Like the attribute values from one database may differ from another database. For example, the date format may differ like“MM/DD/YYYY”or“DD/MM/YYYY”.

3.3 DATA REDUCTION

This process helps in the reduction of the volume of the data which makes the analysis easier yet produces the same or almost the same result. This reduction also helps to reduce storage space. There are some of the techniques in data reduction are Dimensionality reduction, Numerosity reduction, Data compression.

Dimensionality reduction: This process is necessary for real-world applications as the data size is big. In this process, the reduction of random variables or attributes is done so that the dimensionality of the data set can be reduced. Combining and merging the attributes of the data without losing its original characteristics. This also helps in the reduction of storage space and computation time is reduced. When the data is highly dimensional the problem called “Curse of Dimensionality” occurs.

Numerosity Reduction: In this method, the representation of the data is made smaller by reducing the volume. There will not be any loss of data in this reduction.

Data compression: The compressed form of data is called data compression. This compression can be lossless or lossy. When there is no loss of information during compression it is called lossless compression. Whereas lossy compression reduces information, but it removes only the unnecessary information.

3.4 DATA TRANSFORMATION

Data Transformation:

The change made in the format or the structure of the data is called data transformation. This step can be simple or complex based on the requirements. There are some methods in data transformation.

Smoothing: With the help of algorithms, we can remove noise from the dataset and helps in knowing the important features of the dataset. By smoothing we can find even a simple change that helps in prediction.

Aggregation: In this method, the data is stored and presented in the form of a summary. The data set which is from multiple sources is integrated into with data analysis description. This is an important step since the accuracy of the data depends on the quantity and quality of the data. When the quality and the quantity of the data are good the results are more relevant.

Discretization: The continuous data here is split into intervals. Discretization reduces the data size. For example, rather than specifying the class time, we can set an interval like (3 pm-5 pm, 6 pm-8 pm).

Normalization: It is the method of scaling the data so that it can be represented in a smaller range. Example ranging from -1.0 to 1.0.

4. METHODOLOGY

4.1 ARIMA

4.1.1 INTRODUCTION

An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends. A statistical model is autoregressive if it predicts future values based on past values. An ARIMA model can be understood by outlining each of its components as follows:

Autoregression (AR): refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.

Integrated (I): represents the differencing of raw observations to allow for the time series to become stationary (i.e., data values are replaced by the difference between the data values and the previous values).

Moving average (MA): incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations

Each component in ARIMA functions as a parameter with a standard notation. For ARIMA models, a standard notation would be ARIMA with p, d, and q, where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as:

p: the number of lag observations in the model; also known as the lag order.

d: the number of times that the raw observations are differenced; also known as the degree of differencing.

q: the size of the moving average window; also known as the order of the moving average.

4.1.2 HOW ARIMA WORKS

ARIMA is achieved by plugging in time series data for the variable of interest. Statistical software will identify the appropriate number of lags or amount of differencing to be applied to the data and check for stationarity. It will then output the results, which are often interpreted similarly to that of a multiple linear regression model.

4.1.3 ARIMA FORECASTING EQUATION

An ARIMA model can be viewed as a "filter" that tries to separate the signal from the noise, and the signal is then extrapolated into the future to obtain forecasts. The ARIMA forecasting equation for a stationary time series is a *linear* (i.e., regression-type) equation in which the predictors consist of *lags of the dependent* variable and/or *lags of the forecast errors*. That is:

Predicted value of Y = a constant and/or a weighted sum of one or more recent values of Y and/or a weighted sum of one or more recent values of the errors.

The forecasting equation is constructed as follows. First, let y denote the d^{th} difference of Y, which means:

If d=0: $y_t = Y_t$

If d=1: $y_t = Y_t - Y_{t-1}$

If d=2: $y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

Note that the second difference of Y (the d=2 case) is not the difference from 2 periods ago. Rather, it is the *first-*

difference-of-the-first difference, which is the discrete analog of a second derivative, i.e., the local acceleration of the series rather than its local trend.

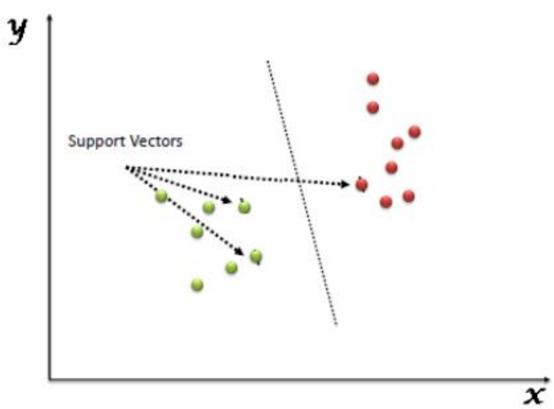
In terms of y, the general forecasting equation is:

$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$

4.2 SUPPORT VECTOR MACHINE

4.2.1 INTRODUCTION

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).



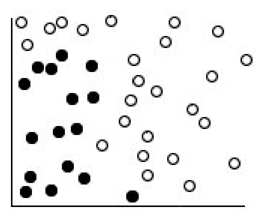
Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

4.2.2 HOW SVM WORKS

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong.

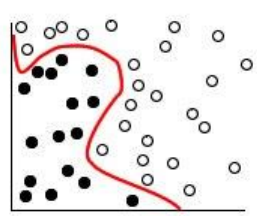
For example, consider the following figure, in which the data points fall into two different categories.

Figure 1. Original dataset



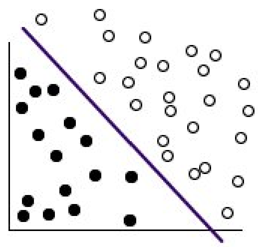
The two categories can be separated with a curve, as shown in the following figure.

Figure 2. Data with separator added



After the transformation, the boundary between the two categories can be defined by a hyperplane, as shown in the following figure.

Figure 3. Transformed data



4.2.3 SVM USING KERNEL FUNCTION

Kernel Function is a method used to take data as input and transform it into the required form of processing data. “Kernel” is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. There are different categories of the kernel functions they are

Gaussian Kernel: It is used to perform transformation when there is no prior knowledge about data.

$$K(x,y) = e^{-(\frac{||x-y||^2}{2\sigma^2})}$$

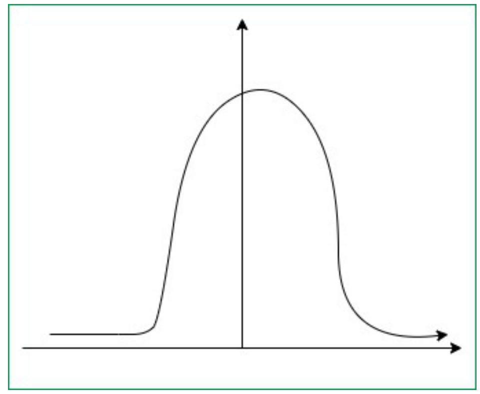
Gaussian Kernel Radial Basis Function (RBF): Same as above kernel function, adding radial basis method to improve the transformation.

$$K(x,y) = e^{-(\gamma||x-y||^2)}$$

$$K(x,x1) + K(x,x2)(Simplified - Formula)$$

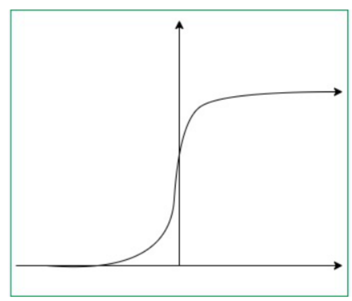
$$K(x,x1) + K(x,x2) > 0(Green)$$

$$K(x,x1) + K(x,x2) = 0(Red)$$



Sigmoid Kernel: this function is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons.

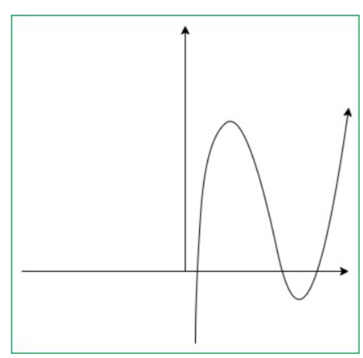
$$K(x,y) = tanh(\gamma.x^Ty + r)$$



Sigmoid Kernel Graph

Polynomial Kernel: It represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel.

$$K(x,y) = tanh(\gamma.x^Ty + r)^d, \gamma > 0$$



Polynomial Kernel Graph

4.3 PROPHET

4.3.1 INTRODUCTION

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. It is accurate and fast and Prophet is used in many applications across Facebook for producing reliable forecasts for planning and goal setting. It is fully automatic and supports Tunable forecasts. Get a reasonable forecast on messy data with no manual effort. Prophet is robust to outliers, missing data, and dramatic changes in your time series.

4.2.2 HOW PROPHET WORKS

It is particularly good at modeling time series that have multiple seasonality and doesn't face some of the above drawbacks of other algorithms. At its core is the sum of three functions of time plus an error term: growthg(t), seasonality s(t), holidays h(t) , and error e_t :

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

g(t): piecewise linear or logistic growth curve for modelling non-periodic changes in time series

s(t): periodic changes (e.g. weekly/yearly seasonality)

h(t): effects of holidays (user provided) with irregular schedules

ε_t: error term accounts for any unusual changes not accommodated by the model

Using time as a regressor, Prophet is trying to fit several linear and nonlinear functions of time as components. Modeling seasonality as an additive component is the same approach taken by exponential smoothing in Holt-Winters technique . We are, in effect, framing the forecasting problem as a curve-fitting exercise rather than looking explicitly at the time-based dependence of each observation within a time series.

At its core, the Prophet procedure is an **additive regression model** with four main components: A piecewise linear or logistic growth curve trend. Prophet automatically detects changes in trends by selecting change points from the data. A yearly seasonal component modeled using Fourier series. A weekly seasonal component using dummy variables. A user-provided list of important holidays.

4.3.2 PROPHET MATHEMATICAL FORM

The Growth Function (and change points):

Changepoints are moments in the data where the data shifts direction. Prophet can automatically detect change points or you can set them yourself. You can also adjust the power the change points have in altering the growth function and the amount of data taken into account in automatic changepoint detection.

The growth function has three main options:

Linear Growth: This is the default setting for Prophet. It uses a set of piecewise linear equations with differing slopes between change points and he slope(m) and offset(b) are variable and will change value at each changepoint.

Logistic Growth the growth term will look similar to a typical equation for a logistic curve (see below), except it the carrying capacity (C) will vary as a function of time and the growth rate (k) and the offset(m) are variable and will change value at each change point.

$$g(t) = \frac{C(t)}{1 + x^{-k(t-m)}}$$

The Seasonality Function: The seasonality function is simply a Fourier Series as a function of time. If you are unfamiliar with Fourier Series, an easy way to think about it is the sum of many successive sines and cosines.

$$s(t) = \sum_{n=1}^{N} (a_n \, cos(\frac{2\pi nt}{P}) + b_n \, sin(\frac{2\pi nt}{P}))$$

The Holiday/Event Function:

The holiday function allows Facebook Prophet to adjust forecasting when a holiday or major event may change the forecast.

References:

D. Shah and K. Zhang, "Bayesian regression and Bitcoin," 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2014, pp. 409-414, doi: 10.1109/ALLERTON.2014.7028484. https://ieeexplore.ieee.org.egateway.chennai.vit.ac.in/document/9452976

Muniye, Temesgen & Rout, Minakhi & Mohanty, Lipika & Satapathy, Suresh. (2020). Bitcoin Price Prediction and Analysis Using Deep Learning Models. 10.1007/978-981-15-5397-4_63.

https://www.researchgate.net/publication/345262720_Bitcoin_Price_Prediction_and_Analysis_Using_Deep_Learning_Models

Cayir, Aykut & Kozan, Ozan & Dağ, Tuğçe & Yenidoğan, Işıl & Arslan, Çiğdem. (2018). Bitcoin Forecasting Using ARIMA and PROPHET. 10.1109/UBMK.2018.8566476.

https://www.researchgate.net/publication/329400738_Bitcoin_Forecasting_Using_ARIMA_and_PROPHET

Ankalkoti, Prashant & Santhosh,. (2017). A Relative Study on Bitcoin Mining. "Imperial Journal of Interdisciplinary Research (IJIR). 3.

https://www.researchgate.net/profile/Prashant-Ankalkoti/publication/318850089_A_Relative_Study_on_Bitcoin_Mining/links/59815084a6fdccb310ff23c8/A-Relative-Study-on-Bitcoin-Mining.pdf

Ji, Suhwan & Kim, Jongmin & Im, Hyeonseung. (2019). A Comparative Study of Bitcoin Price Prediction Using Deep Learning. Mathematics. 7. 898. 10.3390/math7100898.

https://www.researchgate.net/publication/336061476_A_Comparative_Study_of_Bitcoin_Price_Prediction_Using_Deep_Learning