An Empirical Evaluation of Bitcoin Price Prediction Using Time Series Analysis and Roll Over

N. M. Dhanya

Abstract Bitcoin has attracted considerable attention in today's world because of the combination of encryption technology along with the monetary units. For traders, Bitcoin leads to a promising investment because of its highly fluctuating price. Block chain technology assists in the transactions of documentation. The characteristics of the bitcoin which is derived from the blockchain technology has led to diverse interests in the field of economics. The bitcoin data is selected from 2013 to 2018, over a period of 5 years for this analysis. Here a new roll over technology is applied where new data is obtained over time which will close out the old information during machine training. This mechanism will help in incorporating new information in the short-term learning. The results show that the rollover mechanism improves the time series prediction accuracy.

Keywords Bitcoin · Time series analysis · Price prediction · Roll over mechanism

1 Introduction

Time series information in the real world involves the properly ordered and observed sequence of values of any real-world objects, or process or domain information or the values coming straight from the sensor fixed to capture the underlying information. These time series information data come at periodic intervals with the same or different frequencies so that it enables the user to analyse the information from the captured data.

These data therefore offer double advantage to the user, as it helps in past information analysis and also to capture the essence of the future data. Hence modelling of the time series information helps to analyse, predict and solve many prediction problems that might help in various fields of economic growth.

N. M. Dhanya (✉)
Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India
e-mail: nm_dhanya@cb.amrita.edu

327

Hence proper regression algorithms can be employed to analyze the time series data, and to capture the essential information. So, it is necessary to build a proper regression based modeling technique to evaluate the upcoming sensor data. Subsequently, to assess the best model for temperature forecast utilizing the time arrangement information, the model ought to be sufficiently healthy to stay away from commotion; it ought to likewise be exceedingly solid when working with information influenced with expectation inclination or with scaling mistake. The chosen model ought to be adaptable, so that the model can be utilized to contrast the forecast that comes about and distinctive datasets.

1.1 Need for Bitcoin Prediction

Bitcoin has of late pulled in significant consideration in the fields of financial markets, cryptography, and PC science because of its inalienable nature of consolidating encryption, innovation and money related units. Bitcoin uses the Blockchain technology which provides transparency to the transactions made and hence has become a popular means of transaction across the globe. However, the extreme volatility in the Bitcoin values is a reason for concern for investors as well as regulatory authorities. Hence a reliable prediction model for bitcoin price movements is the need of the hour. The observed results is compared with other straight and non-direct benchmark models on predicting bitcoin price. Figure 1 shows the bitcoin price fluctuations from 2012 to 2020 [1].
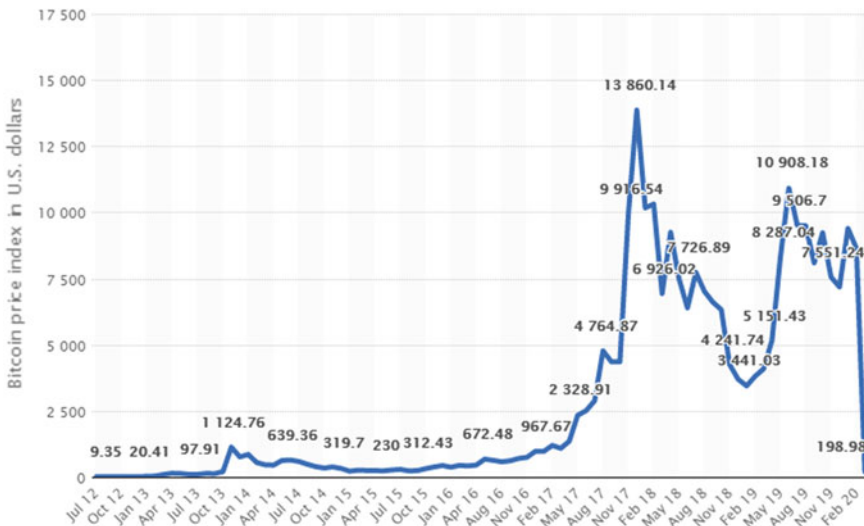

Fig. 1 Bitcoin price changes

Bitcoin is an effective figure cash brought into the money related market in view of its one of a kind convention and Nakamoto's orderly basic detail [2]. Dissimilar to existing fiat monetary standards with national banks, Bitcoin points to accomplish finish decentralization. Inborn attributes of Bitcoin inferred from Blockchain advancements have prompted different research interests in the field of financial aspects as well as in cryptography and machine learning. A machine trained just with Bitcoin value list and changed costs displays poor prescient execution [3]. Our model looks at the precision of anticipating Bitcoin cost through linear Regression, Support Vector Regression, Random forest regression algorithms, ARIMA model and Deep learning approaches.

2   Literature Survey

Existing work on forecasting bitcoin prediction techniques are: Gaussian Process; Linear Regression; Sequential minimal optimization (SMO) regression and Multilayer Perceptron. Gaussian Process [2] implements a classifier function without hyper-parameter tuning for regression. This method however is computationally expensive. Linear regression [4, 5] models the relationship between one or more explanatory variables and a scalar response. It has two main drawbacks i.e., it is limited to linear relationships and it looks for only the dependant variable's mean. Multilayer perceptron [3] is to forecast the webpage views and it makes use of back-propagation method. After the network is built, during the training time it can be monitored and modified. It is quite painful to select the suitable architecture of the network. SMO regression [6] where it has potential for speeding up the forecasting process and also it scales linearly with the size of the training set. But it fails to handle the large-scale training problems because of memory issues. Li and Moore [7] presented an algorithm called Elastic Smooth Season Fitting (ESSF) algorithm which derives the seasonality employing residual sum of squares minimization by smoothness regularization. ESSF accuracy improves significantly over other methods that ignore the yearly seasonality. Jung and Lee [8] created a bitcoin prediction system using block chain information. The authors applied a Bayesian neural network for predicting the price. Li et al. [9] used an LSTM architecture for predicting the Bitcoin price. Block chain statistics is also used for prediction. Various machine learning and deep learning algorithms can be used for prediction [10] and Table 1 shows the functions and limitations of the existing time series models. Sriwiji and Primandari [11] used a Bayesian regularization network for predicting the bitcoin price. They employed a subset selection technology to reduce the number of features and were able to get an accuracy of 91%. Sin and Wang [12] proposed an ANN based ensemble approach based on Genetic algorithm based selective neural network. The next day price of the bitcoin is predicted from the past 50 days observation over 200 features. This strategy generated 85% of the returns.

The following Table 2 shows the already existing time series models for prediction.

Table 1 Functions and limitations of time series models

Models	Function	Cons
ARIMA	Combination of auto regressive, integrated moving average	Noise not considered and parametric
ANN	Linear neuron model, learning by delta rule	Not applicable for non-linear relationships
SVM	Linear learning using decision boundary	Cannot deal with non-linearly separable and noisy data
Recurrent (MLP)	Non linear models, supervised learning by feedback based back propagation	Memory is needed to store temporal information

Table 2 Existing bitcoin prediction techniques

Models	Function
Cryptocurrency price prediction using news and social sentiments [4]	Analyses the ability of news and social media data to predict price fluctuations
Bayesian regression and Bitcoin [6]	Analyses the effect of Bayesian technique for predicting bitcoin fluctuations
Predicting fluctuation in cryptocurrency transactions based on user comments and replies [7]	Trains prediction model based on crawling data from online cryptocurrency communities
Predicting bitcoin price fluctuation with twitter sentiment analysis [14]	Uses sentiment analysis and option mining for prediction
Automated bitcoin trading via machine learning algorithms [15]	Studies the Bayesian neural network algorithm for predicting prices

3 DataSet

For data modelling and bitcoin prediction, time series bitcoin dataset is chosen each with 4000 rows (Time series bitcoin price data from 2011 to 2018) and 24 columns. Preliminary pre-processing of data is done and missing data is filled. This dataset has the following features [13].

- Date: Date of the bitcoin price observation.
- btc_market_price: Average market price of bitcoin.
- btc_total_bitcoins: Total number of bit coins mined.
- btc_market_cap: The total value of the bitcoin in circulation.
- btc_trade_volume: The total value of trading volume of bitcoin.
- btc_blocks_size: Total size of all headers and transation in the block chain.
- btc_avg_block_size: The average block size in MB.
- btc_n_orphaned_blocks: Total number of mined blocks which are not attached to the blockchain.
- btc_n_transactions_per_block: The average number of transactions per block.

- btc_median_confirmation_time: The time for a bitcoin transaction to accept into a mined block.
- btc_hash_rate: The estimated number of tera hashes per second the Bitcoin network is performing.
- btc_difficulty: A measure on the difficulty in finding a new block.
- btc_miners_revenue: The total rewards and fees paid to miners.
- btc_transaction_fees: The total value of all transaction fees paid to miners.
- btc_cost_per_transaction_percent: miners revenue as percentage of the transaction volume.
- btc_cost_per_transaction: miners revenue divided by the number of transactions.
- btc_n_unique_addresses: The total number of unique addresses used on the Bitcoin blockchain.
- btc_n_transactions: The number of daily confirmed Bitcoin transactions.
- btc_n_transactions_total: Total number of transactions.
- btc_n_transactions_excluding_popular: The total number of Bitcoin transactions, excluding the 100 most popular addresses.
- btc_n_transactions_excluding_chains_longer_than_100: The total number of Bitcoin transactions per day excluding long transaction chains.
- btc_output_volume: The total value of all transaction outputs per day.
- btc_estimated_transaction_volume: The total estimated value of transactions on the Bitcoin blockchain.
- btc_estimated_transaction_volume_usd: The estimated transaction value of bitcoin.

A feature engineering mechanism of XGBoost with Bayesian optimization is applied on the features to reduce it and we selected top 15 contributing features. The feature importance is given in Fig. 2.

4   Architecture and Roll Over Mechanism

The below flow chart in Fig. 3 describes the process flow of the proposed work. And Fig. 4 describes the overall workflow of the models applied.

5   Prediction of Error Rate by Rollover

The prediction analysis is done with the traditional models such as linear regression, random forest and SVM and time series models such as ARIMA and LSTM. A roll over mechanism is used to improve the accuracy of the methods. Since the variations in bitcoin price is very high, this rollover mechanism will allow us to iterate our model with latest information in time thus making the model more dynamic and current context aware. The rollover mechanism will work as follows. A time frame
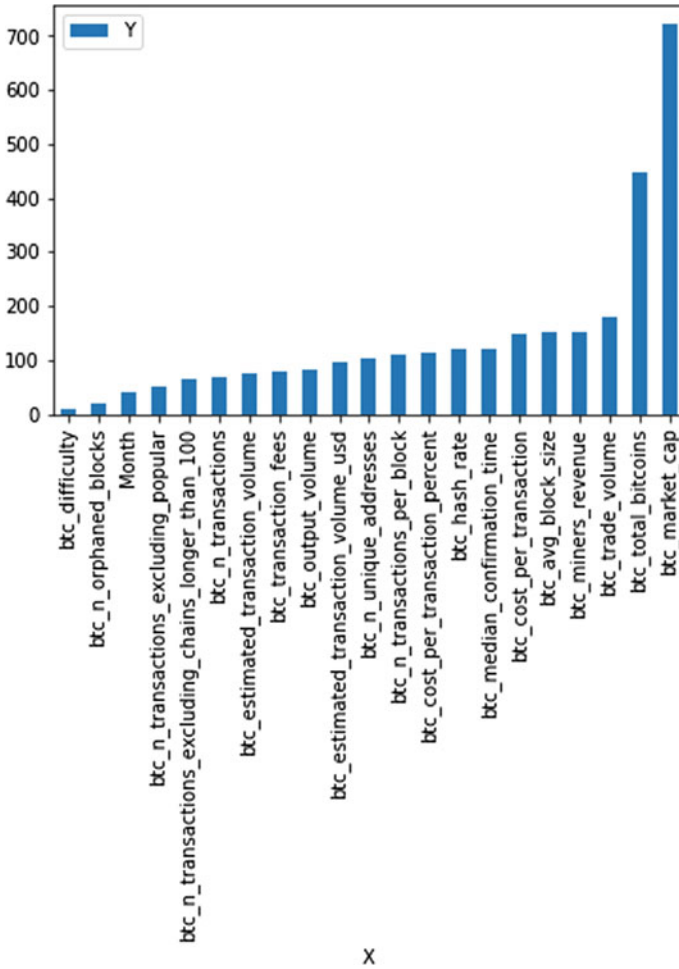
Fig. 2 Feature selection

is set for rolling over in such a way that the old data is closed over time and new data is acquired for rollover. The schematic definition of this method is given in Fig. 5.

The training of the framework will start with N training samples N_{train}, and the prediction performance is tested with the testing data N_{test}. After a time frame of t'-t from time frame t, the model is trained again with a training data N_{train} from time t' and updation to the old model is done. Again, the testing is preformed using test data N_{test}. Similarly training and updation is done through the entire dataset.
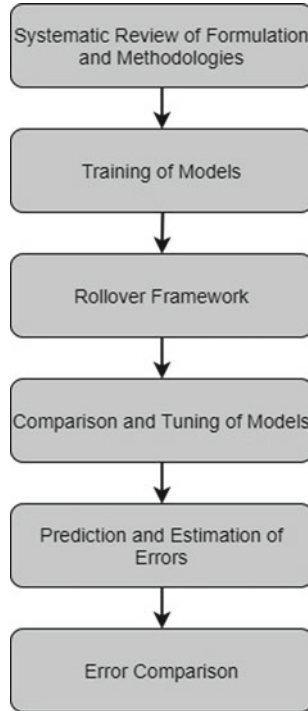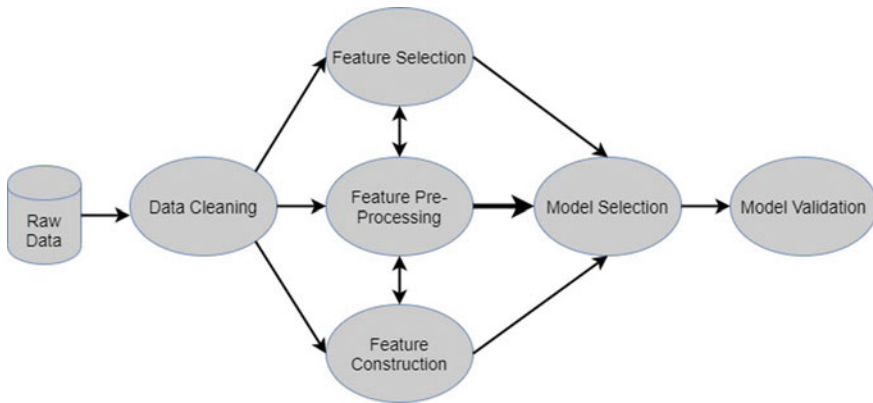
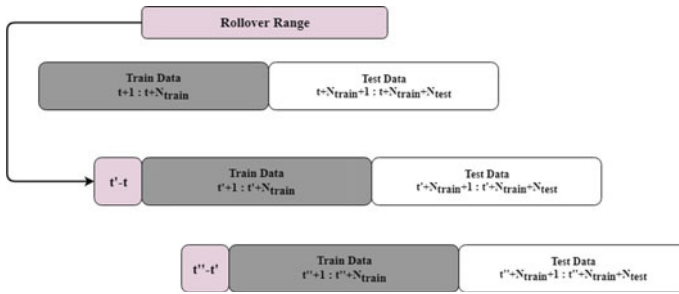**Fig. 3** Overall design

Fig. 4 Prediction model

Fig. 5 Rollover Framework

6 Prediction Using Classical Regression Algorithms

Linear Regression: This model predicts the connection between the indicator and reaction factors utilizing straight indicator capacities which are extricated from the information. The parameters for straight relapse are recognized utilizing relationship, and the components which are much corresponded shapes the indicator factors.

Random Forest Regression: This is an added substance model, where information is anticipated by consolidating the aftereffects of different choices obtained from the easier base models. Henceforth the last yield show is the aggregate of all the less difficult models. This empowers to accomplish preferred outcomes over alternate methods.

Support Vector Regression: It is the enhancement-based relapse system where in each stage, the inclination work relating to the information is improved to yield preferable outcomes over the past stage. Improvement is being performed on the negative angle side where the comparing relapse tree is fitted.

7 Prediction Using Time Series Models

ARIMA is a classic time series analysis algorithm. For applying ARIMA the data should be stationary. From Fig. 6, it can be observed that bitcoin price has an exponential trend, the confirmation of this is one with Augmented Dicky Fuller Test, where the p value is >0.05. Hence, log of the data is taken to make it stationary, but the data appears to be still seasonal and as the last step differencing is applied to remove the trend and seasonality. Figure 4 represents all these transformations. The differencing is done automatically by the ARIMA model.

The ARIMA results are shown in the Fig. 7, which depicts the actual and predicted value of the bitcoin price.

LSTM-based prediction model is created and evaluated in various historical window sizes and network parameters. The window size of 30 was giving the minimum RMSE. A simple LSTM model is created with the following layers and
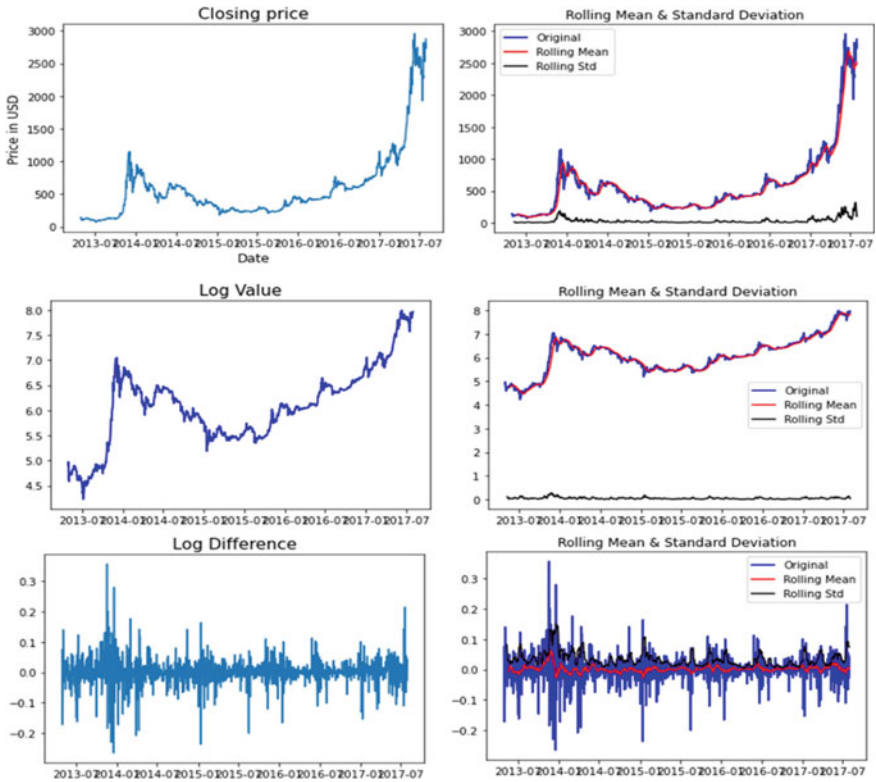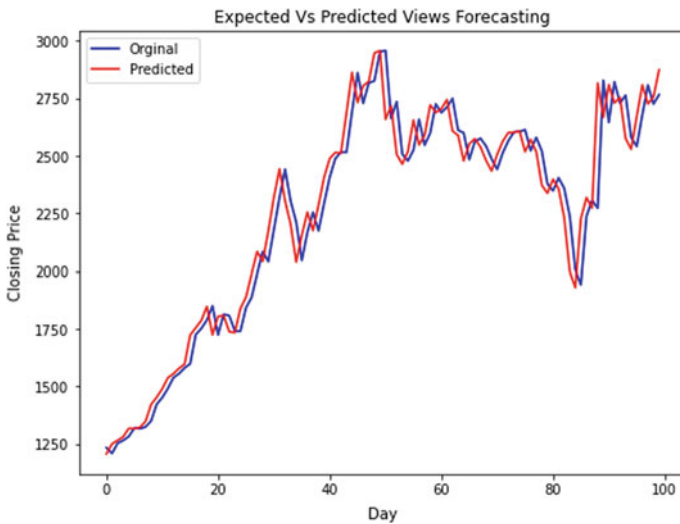
Fig. 6 Original data, log and log-differencing

Fig. 7 ARIMA prediction result

```
Model: "sequential_4"
_____
Layer (type)                  Output Shape              Param #
===============================================================
lstm_5 (LSTM)                 (None, 50)                10400
_____
dense_3 (Dense)               (None, 1)                 51
===============================================================
Total params: 10,451
Trainable params: 10,451
Non-trainable params: 0
_____
```
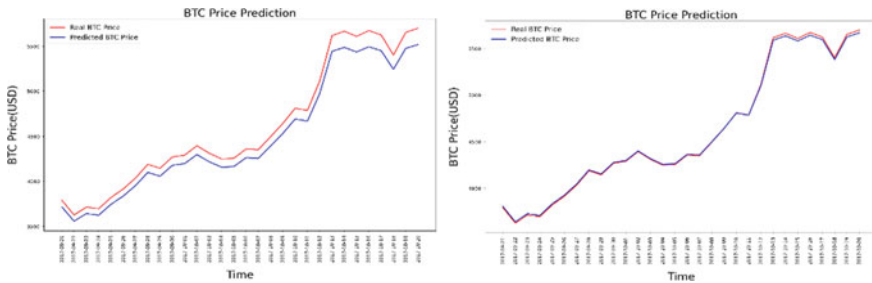
Fig. 8 Neural network model

Fig. 9 LSTM and GRU prediction

the result is shown in Fig. 8. A similar architecture is tried for a GRU also and, it is giving much better prediction as LSTM. Figure 9 describes the results of the deep learning model. As shown in the result, it is clear that GRU gives better prediction accuracy than LSTM model.

8 Performance Metrics for Prediction

Each of the algorithms is tested for the following metrics as a result of rollover framework run.

Mean Squared Error (MSE): It is the metric that is computed by taking the normal distinction between the square of the anticipated qualities and the real qualities. It says how the anticipated qualities are near the relapse line.

$$MSE = \text{Average}(y - y')$$

where, y is the actual values and y' is predicted values.

Root Mean Squared Error (RMS Error): It is characterized as the square foundation of Mean Squared Error. With the end goal of correlation of models, Normalized RMS estimates are utilized here. RMS = (MSE).

$$\text{Normalised RMS} = \frac{\sqrt{\text{MSE}}}{\left(y\text{max} - y\text{min}\right)}$$

Variance: This metric measures how far the qualities got strayed from the mean estimate.

9 Methodology and Execution

- Choosing the Data record: The dataset with 15 features and 4000 rows fills in as the contribution for the accuracy calculation from CSV document.
- Dependent and the independent factors are appropriately inputted to the model.
- Generation of Training and Testing information: The given dataset is 10 fold cross-validated to get the preparation and testing information with the proportion of 80:20.
- Model forecast: The Selected information is tested with the various models.
- Each calculation is executed in python where the parameters for the models are kept steady for all the datasets.
- For SVM regression, number of estimators is settled at 25, minimum leaves is set to 5, and number of arbitrary states is set at 3.
- For Random forest regression, number of estimators is settled at 25, learning rate is set to 0.2, maximum profundity is set at 5 and number of irregular states is set to 3.
- ARIMA is applied with $p, d, q = (2, 1, 0)$.
- LSTM and GRU is tried with 50 nodes and for 100 epochs.
- Then the algorithm is adjusted for rollover framework with all models accessible and the parameters of the ideal model is settled.
- The above model setting is rehashed for all the run and the measurements estimated.

10 Results and Discussions

The following Table 3 shows the execution results of classic algorithms run on the dataset.

From Table 3 results, application of Rollover Framework using deep learning techniques executes better than the other classical techniques ultimately resulting in minimization of error rate.

Table 3 Execution results

Models	Metric	Without rollover	With rollover
Linear	RMSE	0.4823	0.3863
	Variance	0.2663	0.2655
Random forest	RMSE	0.0039	0.5039
	Variance	0.0138	0.0124
SVM	RMSE	0.3201	0.2801
	Variance	0.0428	0.0098
ARIMA	RMSE	0.0875	0.0359
	Variance	0.9314	0.9445
LSTM	RMSE	0.0825	0.0325
	Variance	0.9147	0.9436
GRU	RMSE	0.0725	0.0315
	Variance	0.9185	0.9523

11 Conclusion

Bitcoin is a cryptocurrency mechanism which is extensively studied. In this paper, the Bitcoin prize is analysed using time series analysis. A linear model, random forest and SVM is applied and the results are analysed. The GRU model with Rollover is giving the highest accuracy in predicting the closing price. The LSTM and GRU model can be improved by hyperparameter tuning such as dropout and other regularization techniques.

References

1. https://www.statista.com/statistics/326707/bitcoin-price-index/
2. Lane ND, Bhattacharya S, Georgiev P, Forlivesi C, Kawsar F (2015) An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices. In: Proceedings of the 2015 international workshop on internet of things towards applications—IoT-App 15
3. Spuler M, Sarasola-Sanz A, Birbaumer N, Rosenstiel W, Ramos-Murguialday A (2015) Comparing metrics to evaluate performance of regression methods for decoding of neural signals. In: 37th Annual international conference of the IEEE engineering in medicine and biology society (EMBC)
4. Ahmed NK, Atiya AF, Gayar NE, El-Shishiny H (2015) An empirical comparison of machine learning models for time series forecasting. Technical Report
5. Veerakumar S, Dhanya NM (2018) Performance analysis of various regression algorithms for time series temperature prediction. J Adv Res Dyn Control Syst 10(3):996–1000
6. Anufriev M, Hommes C, Makarewicz T (2012) Learning to forecast with genetic algorithms. Working Paper
7. Khadka M, Popp B, George KM, Park N (2010) A new approach for time series forecasting based on genetic algorithm. In: CAINE, pp 226–231

8. Jang H, Lee J (2018) An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. IEEE Access 6:5427–5437
9. Li L, Arab A, Liu J, Liu J, Han Z (2019) Bitcoin options pricing using LSTM-based prediction model and blockchain statistics. In: 2019 IEEE international conference on Blockchain (Blockchain)
10. Dhanya NM, Harish UC (2018) Sentiment analysis of twitter data on demonetization using machine learning techniques. In: Lecture notes in computational vision and biomechanics, vol 28, pp 227–237
11. Sriwiji R, Primandari AH (2020) An empirical study in forecasting bitcoin price using bayesian regularization neural network. In: Proceedings of the 1st international conference on statistics and analytics, ICSA 2019, 2–3 Aug 2019, Bogor, Indonesia
12. Sin E, Wang L (2017) Bitcoin price prediction using ensembles of neural networks. In: 2017 13th International conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD)
13. Kaggle. https://www.kaggle.com/c/bitcoin/data
14. Dobslaw F (2010) A parameter tuning framework for metaheuristics based on design of experiments and artifcial neural networks. In: Proceedings of the international conference on computer mathematics and natural computing (WASET '10)
15. https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice, http://scikit-learn.org/stable/modules/cross_validation