Journal of Vol.3 , No.1 , PP.70-87 , 2024

Bitcoin_ML: An Efficient Framework for Bitcoin Price Prediction Using Machine Learning

Maged Farouk[a] , Nashwa S Ragab[a], Diaa Salama[b,c,*], Omnia Elrashidy[a], Lana Mandour[a], Mariam Ahmed[a], Jana Walid[a], Malak Mesbah[a], Rawan Attia[a], Nouran Ahmed[a], Reda Elazab[a]

[a]*Department of Business Information Systems, Faculty of Business, Alamein International University, Alamein, Egypt*

[b] *Faculty of Computers Science, Misr International University, Cairo, Egypt*

[c] *Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt*

[*]Corresponding Author: Diaa Salama [***diaa.salama@miuegypt.edu.eg***]

ARTICLE DATA

Article history:
Received 06 Jan 2024
Revised 13 Jan 2024
Accepted 02 Feb 2024
Available online

Keywords:
Bitcoin Price Prediction
Machine Learning
Artificial Intelligence
Algorithm
Linear Regression
Random Forest

ABSTRACT

Econometrics can be used to understand and forecast price movements, assess market efficiency, and explore the factors influencing Bitcoin's value and adaptation.
Econometrics is related to bitcoin in seven categories: price analysis and prediction, market efficiency, determination of Bitcoin prices, risk analysis, adaptation and network effects, causality tests, and simulation and stress. Testing these analyses can be invaluable for policymakers, investors, and financial institutions interested in the economics of digital currencies.
Bitcoin price prediction in machine learning has many challenges that have deep roots in 2 main properties: cryptocurrencies and complexities in the Machine Learning models.
Many problems are associated with machine learning for bitcoin price prediction, such as overfitting, data quality and availability, latent variables, model interpretability, computational complexity, dynamic adaptation, market manipulation, anomalies, data snooping bias risk, and time horizon mismatch. In the paper, we proposed an efficient framework for the prediction of bitcoin using nine different machine learning algorithms (linear Regression, random forest, adaboost, tree, KNN, gradient boosting, constant, neural network, SVM) on five different datasets. The results revealed that linear Regression emerged as the optimal model for the first data set. In the second data set, the random forest model demonstrated superior performance. The third data set exhibited the highest efficacy when the Adaboost model was employed. The fourth data set yielded the best outcomes with the random forest model, while linear Regression was the most effective choice for the final data set.

1. Introduction

Econometrics uses observed data to study economic phenomena systematically. The primary rule of Econometrics is to test. Fortunately, this rule has been used in experimental applications easily and accurately. Econometrics also provides quantitative estimates of price and income elasticities of demand, the efficiency of production processes as captured by the cost function, etc. These are important for policy decision-making [1-3].

We can consider predicting cryptocurrency's price as a common type of time series problem, such as predicting the price of Bitcoin. Traditional time-serious methods, such as the well-known AutoRegressive Integrated Moving Average (ARIMA) model, have been applied to predict cryptocurrency price and movement. One of the main problems with Bitcoin price predictions is that they lack sufficient analytical support to back up their claims [4-6].

Machine learning is the field of developing computer algorithms capable of imitating human intelligence. It relies on ideas from Artificial intelligence, probability, and statistics. One of the most important characteristics of these algorithms is their distinctive ability to learn the data landscape from the

input data, with or without knowledge. By expertly deciphering hidden patterns within complex data, machine learning algorithms can create accurate models of Bitcoin price predictions [7-9].

Machine learning methods can predict Bitcoin price at different frequencies, corresponding to how the price is measured. Extensive data points are used to predict the daily price of Bitcoin, while the technical trading features derived from cryptocurrency exchanges are used for shorter-term predictions. Predicting the prices of established financial markets, like Bitcoin prices, has traditionally relied on methods like Holt-Winter models. These models thrive on stable trends, predictable cyclical fluctuations, and low variance; however, these assumptions Show cracks in the face of highly unpredictable markets like Bitcoin. It is similar to trying to predict prices with no clear patterns, only an unpredictable fluctuation [10-12].

The main contribution of this paper can be summarized as follows: Investigating the efficacy of machine learning for Bitcoin price prediction. Evaluating nine machine learning algorithms (linear Regression, random forest, AdaBoost, decision tree, KNN, gradient boosting, constant, artificial neural network, and SVM) across five different datasets.

The rest of the paper is organized as follows: related work is mentioned in the third section, which is the section below. Moving on to the Methodology in the fourth section, we discuss the datasets, the algorithms used, and the performance metrics. Subsequently, we show the results in the fifth section where we determine the best and worst models for the datasets discussed in the Methodology. The conclusion is located in the sixth section, where we provide a summarized statement based on the work done in the paper. Finally, an acknowledgment can be found in the seventh and final section.

3. Related Work

In[13], This research uses machine learning models to predict Bitcoin's USD price movement. They use Bayesian-optimized recurrent neural networks (RNN) and Long Short Term Memory (LSTM) networks, achieving 52% accuracy and 8% error. Comparatively, the ARIMA model performs poorly. The study addresses the scarcity of machine learning studies in Bitcoin prediction, comparing traditional and modern approaches using Bitcoin data and Blockchain information. Prior studies tried different methods, but some faced limitations due to small sample sizes and social media influence. The paper follows a specific research method, aiming to accurately predict Bitcoin's price changes using deep learning models, offering insights into a less-explored area in Bitcoin price forecasting.

The research paper [14] states that, in the domain of Bitcoin Price Prediction using machine learning models, it became evident that the efficiency of RNN, specifically when equipped with LSTM. Encompassing the data sets from 2012 to 2019, a comprehensive evaluation was conducted based on robust metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R squared. The test of RNN, specifically with short-long memory (LSTM), comes to the fore as notably more proficient in discerning protracted dependencies for accurate Bitcoin Price Predictions. A thorough examination of RMSE, MAE, and R-squared metrics distinctly underscores the highlighted accuracy of RNN over the linear regression model. This has been reaffirmed through graphical representations, delineating actual Bitcoin prices against those predicted by the RNN model.

As stated in [15], the blockchain is a public ledger that records all Bitcoin transactions, with new blocks being added every 10 minutes via mining. Because of their ability to analyze massive datasets and detect trends, machine learning techniques have been widely employed for Bitcoin price prediction. Linear regression, support vector machines (SVM), long short-term memory (LSTM) networks, random forest, and deep learning are examples of commonly used methods. These methods are designed to provide more accurate predictions about future price changes, benefiting individual investors and financial institutions active in bitcoin markets. Because of Bitcoin's popularity and volatility, studying Bitcoin price prediction using machine learning techniques is an important research topic.

The research paper [16] focuses on predicting Bitcoin prices using machine learning models, specifically random forest regression and LSTM algorithms. The study collected data from 7 natural years, from March

31, 2015, to April 1, 2022, and analyzed the impact of Bitcoin's price bubbles in 2017 and 2021 on prediction accuracy. The research used 47 explanatory variables across various categories, including cryptocurrencies, commodities, market indices, foreign exchange, and public attention variables. The results show that the random forest regression model outperformed the LSTM algorithm regarding prediction accuracy, with lower RMSE and MAPE. The study also identified the importance of variables in predicting Bitcoin prices, with the OHLC price of Bitcoin in the previous period ranking high and the importance of certain variables changing over time. The research paper focuses on using machine learning models to predict Bitcoin prices, analyzing the impact of explanatory variables, the importance of periods, and the accuracy of different models in predicting Bitcoin prices. The authors used a methodology that involved predicting Bitcoin prices using machine learning models, specifically random forest regression and LSTM algorithms. They conducted a comprehensive analysis of the existing methodologies and models for predicting cryptocurrency prices, discussed the setting of model parameters and error settings, analyzed the selection and pre-processing of explanatory variables, evaluated the performance of the proposed model, and discussed the limitations of the research and directions for future attempts. The study also collected data from various sources, dividing the data into two periods for independent research, training models for their respective periods, and conducting experiments and attempts to select explanatory variables for each period. Moreover, the authors compared the prediction accuracy of various models and conducted hypothesis tests on the significant differences between different algorithms. The Methodology also included an in-depth analysis of the relationship between model accuracy and the lag of explanatory variables.

Many studies have investigated applying machine learning techniques to predict the price of Bitcoin. A comprehensive investigation was conducted using various regression models, including linear Regression [17], support vector regression, and random forest, and achieved highly impressive accuracy. They used a 1-minute interval trading data from the Bitstamp exchange between 2012 to 2018, leveraging sci-kit-learn and Keras libraries to implement their best-performing model, attained a Mean Squared Error (MSE) of 0.00002 and an R-Square (R2) of 99.2%, a testament to the power of machine learning in this area. However, their study was limited to data before 2018 and did not explore the impact of news sentiment on price forecasting. Building on this work, our research aims to use newer data, including news sentiment analysis, to develop more accurate and robust models for Bitcoin price prediction.

In [18], authors proposed a two-stage framework utilizing Bayesian Regression and artificial neural networks. Their approach collected historical data and updated a probabilistic model based on identified patterns, incorporating uncertainty and previous knowledge. The data used in their approach was then segmented into three stages: weight calculation, evaluation, and daily price change prediction. While demonstrating the potential of machine learning in this domain, their work focused primarily on linear models. It did not fully explore the capabilities of advanced techniques such as random forests or generalized linear models (GLMs) known for their flexibility and robustness.

4. Methodology

This section demonstrates the testing and scoring of different datasets for Bitcoin price prediction. First, the datasets are tested on several folds and random samples using different algorithms and then compared based on several errors. The best Algorithm is determined based on the least number of errors. Each Algorithm used is explained thoroughly with figures and graphs for a clearer overview.

4.1 Dataset Description

The first Dataset shows the change in bitcoin price from 2014 to 2021, and it consists of six features, namely, Date, representing the day on which the data is recorded. Currency represents the currency the Dataset works on, BTC, short for Bitcoin. The closing price represents the last price that was recorded that day. 24 open contains opening exchange rate daily. 24 high contains information about when the price was

high daily. Finally, 24 low contains information about when the price was low daily. It is important to note that the target is Closing Price.

Table I
Bitcoin Bull-Run Prediction Dataset

Feature	Type	Values
Date	Date	From 2014-03-14 to 2021-10-29
Currency	Categorical	BTC
Closing Price	Numerical	From 110 to 63300
24 open	Numerical	From 110 to 63600
24 high	Numerical	From 120 to 64800
24 low	Numerical	From 84.3 to 62100

The second Dataset shows the change in bitcoin price throughout the years from 2014 to 2022, and it consists of six features namely, Date, representing the day the transaction is made. Open represents the price at the beginning of the day. High shows the day's highest price, while Low shows the lowest price the BTC reached that day. Close displays the closing price, the day's last price, without further changes or adjustments. Finally, Volume shows how many activities happened that day regarding buying, selling, and trading BTC. The target is Close as in the closing price.

Table II
Bitcoin Price Dataset

Feature	Datatype	Values
Date	Date	From 2014-09-17 to 2022-02-19
Open	Numerical	From 177 to 67500
High	Numerical	From 212 to 68800
Low	Numerical	From 172 to 66400
Close	Numerical	From 178 to 67600
Volume	Numerical	From 5.91 million to 351 billion

The third Dataset shows the change in bitcoin price beginning in 2017 and ending in 2020. The BitCoin data is limited to these three years because Dec 2017 was when BitCoin Prices Skyrocketed. Hence, this duration is Perfect for Predicting future Prices. The years before 2017 had a low Price ratio, which can cause a disturbance in our Prediction Models.
The Dataset consists of six features, namely, Date as in the day of the transaction; High, as in the highest price reached that day, as in the lowest price reached that day, as in the price at the beginning of the day; Close as in the price at the end of the day, and Volume as in the number of buying, selling, and trading that took place that day. The target is Close, being the closing price.

Table III
BitCoin Dataset

Feature	Type	Values
Date	Date	From 2017-07-10 to 2020-07-10
High	Numerical	From 2060 to 20100
Low	Numerical	From 1840 to 19000
Open	Numerical	From 1930 to 19500
Close	Numerical	From 1930 to 19500
Volume	Numerical	From 706 million to 74.2 billion

This Dataset shows how the Bitcoin price changes throughout the eight years from 2014 to 2022. It consists of six features: Date, which represents the trading date. Open represents the price of BTC when the market opened. High represents the highest achieved price of the day. Low represents the lowest price of the day. Close is the price of BTC when the market closes. Finally, the Volume shows How much buying or selling happened or how much trade took place. The target is Close, being the closing price.

Table IV
Analyzing and Predicting Bitcoin pricing trend Dataset

Features	Type	Values
Date	Date	From 2014-09-17 to 2022-05-05
Open	Numerical	From 177 to 67500
High	Numerical	From 212 to 68800
Low	Numerical	From 172 to 66400
Close	Numerical	From 178 to 67600
Volume	Numerical	From 5.9 million to 351 billion

The fifth and final Dataset comprises historical records of Bitcoin's price movements over time, capturing daily fluctuations. It includes features such as the Date of the transactions, the currency, which is limited to BTC only, the closing price at the end of the day, the opening price at the beginning of the day, and the high and low representing the highest and lowest achieved prices of the BTC that day. The Dataset spans a considerable timeframe of nine years, offering insights into various market conditions, trends, and potential correlations with external factors, providing a rich source for training models to forecast Bitcoin's future price changes. The target is the Closing price.

Table V
Bitcoin_Prediction_Dataset

Features	Type	Values
Date	Date	From 2013-10-01 to 2021-05-18
Currency	Categorical	BTC
Closing Price	Numerical	From 109 to 63300
24 Open	Numerical	From 109 to 63600
24 High	Numerical	From 119 to 64800
24 Low	Numerical	From 83.3 to 62100

4.2 Algorithms used:

1) Linear Regression provides a linear relationship between an independent and dependent variable to predict the outcome of future events [19].
   The formula of linear Regression is:

$$Y = mx + b \qquad (1)$$

   Where: Y = dependent X = independent m = slope b = intercept

2) Random forest is a tree-shaped diagram that combines the output into many decision trees and turns it into a single result. This Algorithm builds numerous decision trees during the training phase. Each

tree is trained on a subset of the Dataset (randomly sampled with replacement), and at each node of the tree, it considers a subset of features (also randomly selected).
It uses a technique called bootstrapping, which randomly samples the data with replacement. This creates different subsets of the Dataset for each tree, ensuring diversity among the trees [20].

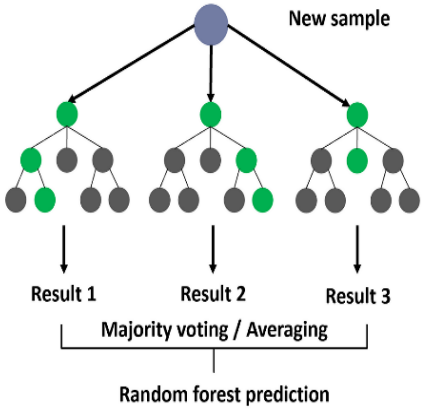


Figure 2. Random Forest Algorithm [20]

3) Adaboost: is short for Adaptive Boosting, a Boosting technique that can be used to classify a large amount of data by combining multiple weak or base learners. It works by weighting the instances in the training dataset based on the accuracy of previous classifications [21].

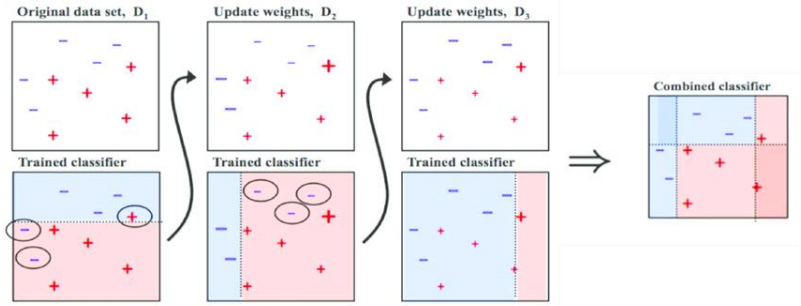


Figure 3. Adaboost Algorithm [21]

The simple weight formula:

$$w(x_i y_i) = \frac{1}{N}, i = 1,2, \dots n \qquad (2)$$

4) Tree: it is called a tree because it has a root, looks like an upside-down tree, and the branches represent the various outcomes [22].
The tree formula:

Expected value (EV) = (First possible outcome x Likelihood of outcome) + (Second possible outcome x Likelihood of outcome) – Cost

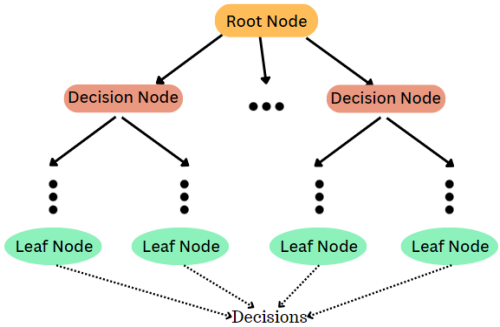


Figure 4. Tree Algorithm [22]

5) Gradient boosting is known for its prediction speed and accuracy, particularly with large and complex datasets. The main idea behind this Algorithm is to build models sequentially, and these subsequent models try to reduce the errors of the previous model. This is done by building a new model on the errors or residuals of the previous model [23].

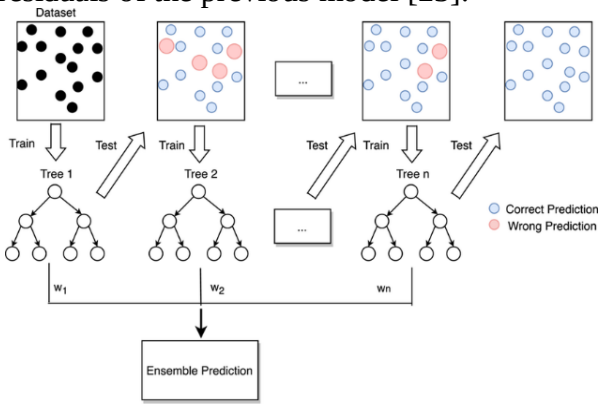


Figure 5. Gradient boosting Algorithm [23]

6) Support Vector Machine (SVM) is a supervised machine learning algorithm for classification and Regression. It is a supervised machine-learning problem where we try to find a hyperplane that best separates the two classes. SVM finds the maximum margin between the hyperplanes, which means the maximum distances between the two classes [24].

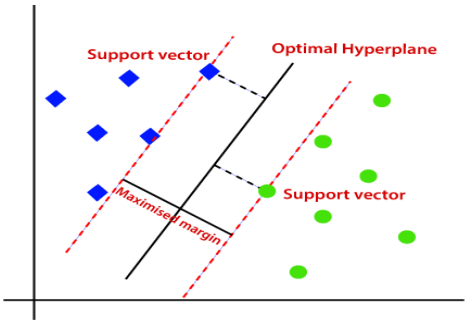


Figure 6. SVM Algorithm [24]

7) The K-Nearest Neighbor (KNN) algorithm is an instance-based or lazy learning algorithm. It doesn't build a model during training but memorizes the entire Dataset. It can be used for classification and regression tasks. It relies on the idea that similar data points tend to have similar labels or values [25].

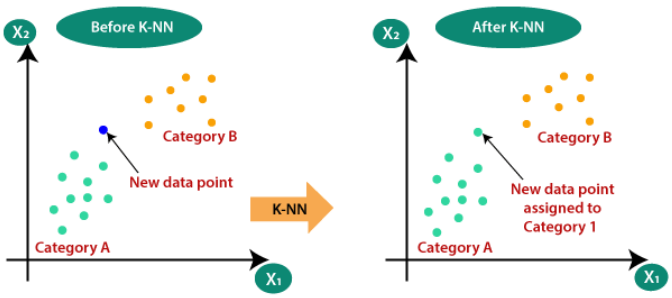


Figure 7. kNN Algorithm [25]

8) Neural networks mimic the fundamental workings of the human brain and draw inspiration from the brain's information processing. They address a wide range of real-time tasks owing to their capacity for rapid computation and quick responses. The remarkable performance of neural networks stems from their proficiency in data-driven learning and predictive decision-making [26].

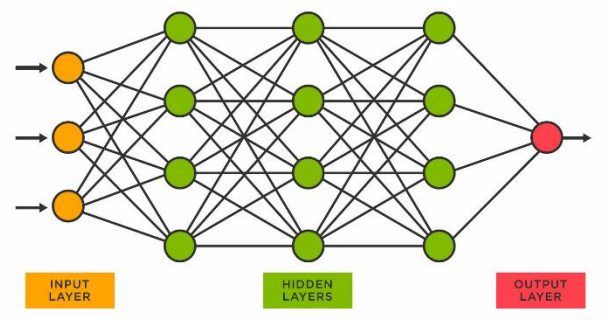
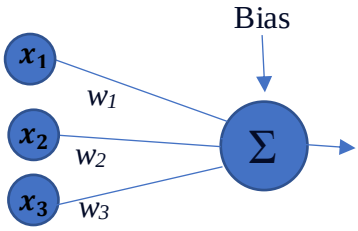


Figure 8. Neural Network Algorithm [26]



$$\text{Activation } \left(\sum_{i=1}^{m} (w_i * x_i) + bias\right) \quad (3)$$

9) Constant refers to a fixed value that does not change during the learning or prediction. Predict the most frequent class or mean value from the training set.

Constants are commonly used in various aspects of machine learning, including model training, feature engineering, and algorithm design. The constant probabilities are the observed priors [27].

4.3 Performance Metrics

Performance Metrics reveal how many errors each model or Algorithm makes in the testing and scoring phase of the Dataset. Two performance metrics were used for the five Datasets we provided, namely, R2 and MAPE. Those are explained in more detail below.

1) R-squared (R2) or the coefficient of determination is a statistical measure in a regression model to ascertain the proportion of variance in the dependent variable, which is an independent variable. We used R-squared to know how well the data fit the regression model. R-squared can take any value between 0 to 1. In addition, it does not mean the correctness of the regression model[28-29].
Calculation of the r-squared :

$$R - Squared = \frac{SSregression}{SStotal} \qquad (4)$$

SS regression is the sum of squares of Regression, and SStotal is the sum of the total sum of squares.

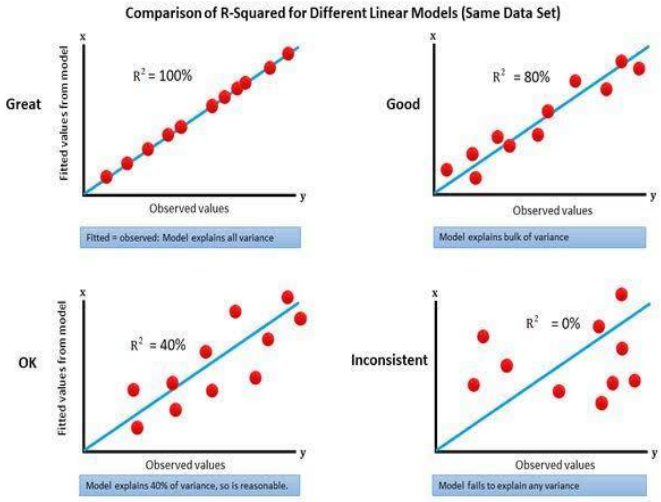


Figure 9. Comparison of R2 Values for different linear models

2) MAPE, or Mean Absolute Percentage Error, is a metric used to measure the accuracy of forecasting methods. It calculates the average of absolute percentage errors for each entry in a dataset, providing insight into how accurate forecasted quantities are compared to actual quantities. A lower MAPE indicates better accuracy. Accurate forecasting can lead to better decision-making, cost adjustments, and alignment of production operations with customer demands. Forecast error measures the deviation between actual and forecasted quantities, focusing on the magnitude of the error rather than its direction. MAPE can be calculated by organizing data and calculating the absolute percent error for each data entry using the formula: Absolute percent error = [(actual-forecast |)/ actual |] x 100.
Then, all the absolute percent errors are added together, and the sum is divided by the number of errors[30]

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \qquad (5)$$

M = mean absolute percentage error

n = number of times the summation iteration happens
A_t = actual value
F_t = forecast value

5. Results and Discussion

Testing and scoring results:

The Dataset is tested two times, once on cross-validation with number of folds of 10 and on the random sample with training 80% and testing 20%. Nine different algorithms are used and compared based on number of errors using MAPE and R2 as references. Comparison is based on MAPE.

The following table and graph for the first Dataset represent the testing and scoring results of the Dataset on cross-validation with number of folds 10. The number of errors for each Algorithm varies from 0.014 being the minimum to 22.149 being the maximum. The best-performing algorithms are Linear Regression, Random Forest, and Adaboost, with Linear Regression being the leading Algorithm with minimum errors. The worst-performing Algorithms are SVM, Neural Network, and Constant with SVM being the Algorithm with the most errors. We can conclude that, in this case, the best Algorithm is Linear Regression, and the worst Algorithm is SVM.

Table VI
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	MAPE	R2
Linear Regression	0.014	1.000
Gradient Boosting	0.028	0.999
Random Forest	0.015	0.999
AdaBoost	0.016	0.999
Tree	0.019	0.998
KNN	0.024	0.998
Neural Network	7.427	-0.001
Constant	5.563	-0.086
SVM	22.149	-1.211

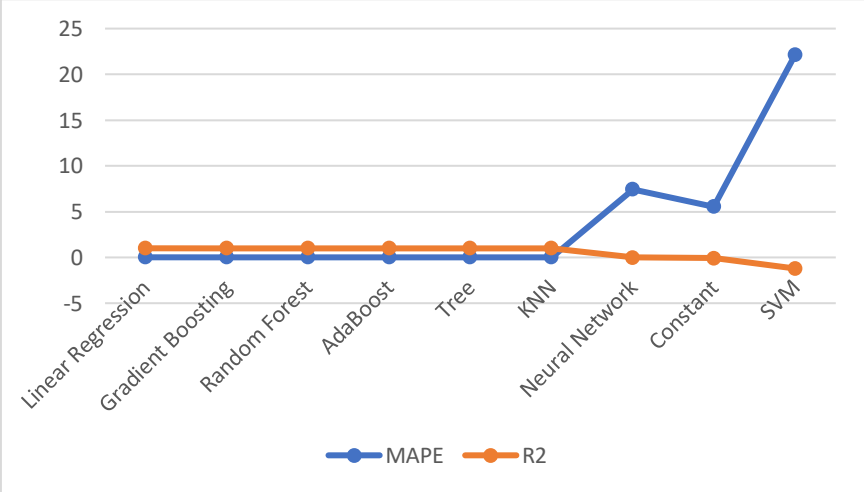


Figure 10. the performance results for the First Dataset p with 10 k-fold

The table and graph below represent the testing and scoring results of the Dataset on the random sample with training of 80% and testing of 20%. The number of errors for each Algorithm varies between 0.014 as the minimum and 20.267 as the maximum. The best-performing algorithms are Linear Regression, Random

Forest and Adaboost with Linear Regression being the leading Algorithm with minimum errors. The worst-performing Algorithms are SVM, Neural Network and Constant with SVM being the Algorithm with the most errors. We can conclude that in this case, the best Algorithm is Linear Regression and the worst Algorithm is SVM.

Table VII
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	MAPE	R2
Linear Regression	0.014	1.000
Gradient Boosting	0.027	0.999
Random Forest	0.015	0.999
AdaBoost	0.017	0.999
Tree	0.019	0.998
KNN	0.026	0.998
Neural Network	7.517	-0.001
Constant	6.559	-0.027
SVM	20.267	-0.999

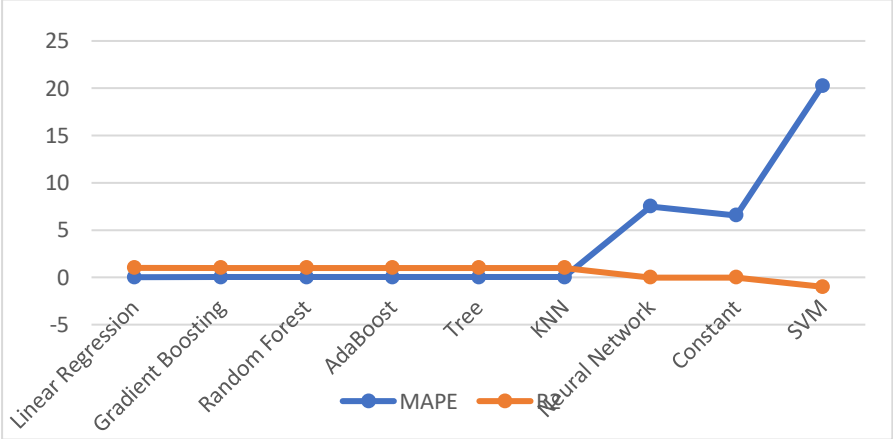


Figure 11. the performance results for the First Dataset with an 80/20 data split

The table and graph below for the second Dataset represent the testing and scoring results of the Dataset on cross-validation with the number of folds 10. The number of errors for each Algorithm varies between 0.012, the minimum, and 26.370, the maximum. The best-performing algorithms are Linear Regression, Random Forest, and Adaboost, with Linear Regression being the leading Algorithm with minimum errors. The worst-performing Algorithms are SVM, Neural Network, and Constant, with SVM being the Algorithm with the most errors. Linear Regression is the best Algorithm, while the SVM remains the worst Algorithm in this Dataset.

Table VIII
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	MAPE	R2
Tree	0.018	0.999
SVM	26.370	-0.989
Random Forest	0.015	0.999
Neural Network	7.305	-0.031
Linear Regression	0.012	1.000
kNN	0.418	0.564
Gradient boosting	0.030	0.999
Constant	10.550	-0.001
AdaBoost	0.016	0.999

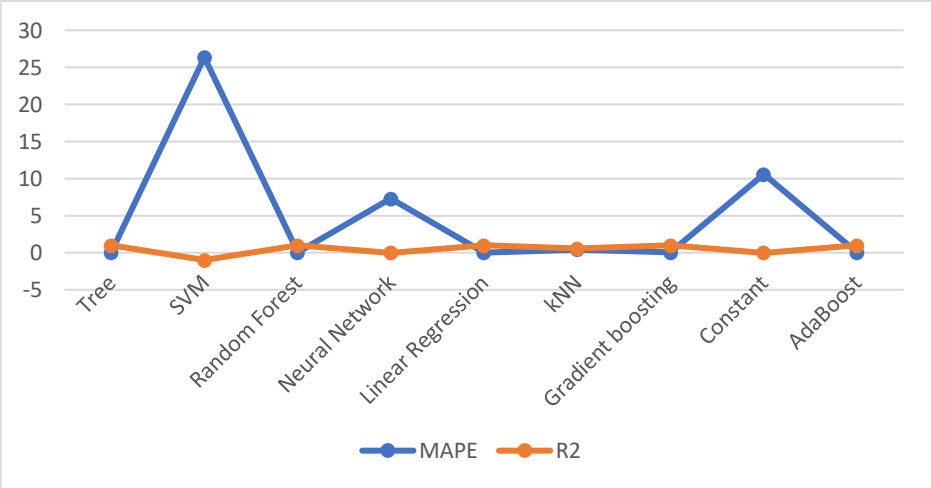


Figure 12. The second dataset performance chart with ten k-fold

The following table and graph show the testing and scoring results of the Dataset on the random sample with training of 80% and testing of 20%. The errors for each Algorithm are very between 0.012 being the minimum and 25.313 being the maximum. Linear Regression, Random Forest, and Adaboost are the top three best-performing algorithms, with Linear Regression also being the model with minimum errors. The worst performing algorithms, however, are SVM, Constant, and Neural Network, with SVM remaining as the Algorithm with the most errors throughout the testing. Based on the previous results, we can conclude that the best Algorithm is Linear Regression, and the worst Algorithm is SVM.

Table IX
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	MAPE	R2
Tree	0.019	0.999
SVM	25.313	-0.832
Random Forest	0.015	0.999
Neural Network	10.356	0.047
Linear Regression	0.012	1.000
kNN	0.421	0.565
Gradient boosting	0.030	0.999
Constant	10.561	-0.000
AdaBoost	0.017	0.999

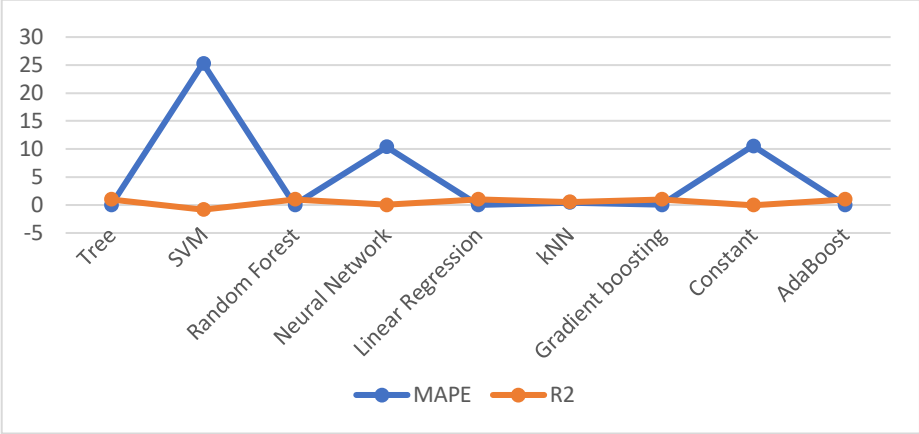


Figure 13. Second Dataset Performance chart with 80/20 data split

The following table and graph for the third Dataset represent the testing and scoring results of the Dataset on cross-validation with the number of folds 10. The number of errors for each Algorithm varies from 0.000 being the minimum to 0.379 being the maximum. The best-performing algorithms are Linear Regression, Random Forest, and Adaboost, with Linear Regression being the leading Algorithm with minimum errors. The worst-performing Algorithms are SVM, Neural Network, and Constant, with constant being the Algorithm with the most errors. We can conclude that, in this case, the best Algorithm is Linear Regression, and the worst Algorithm is constant.

Table X
STATISTICS OF ALGORITHM WITH 10 K-FOLD

Model	MAPE	R2
Linear Regression	0.000	1.000
Gradient Boosting	0.004	1.000
AdaBoost	0.002	1.000
Random Forest	0.003	0.999
Tree	0.004	0.999
KNN	0.255	0.318
Neural Network	0.370	-0.004
Constant	0.379	-0.013
SVM	0.355	-0.016

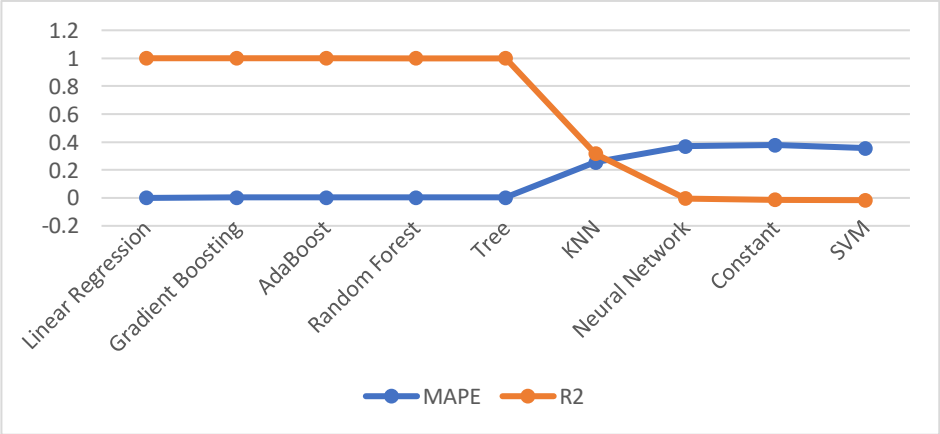


Figure 14. Third Dataset Performance chart using 10 k-fold

The following table and graph represent the Dataset testing and scoring results on random samples with training at 80% and testing at 20%. The number of errors for each Algorithm varies from 0.000 being the minimum to 0.933 being the maximum. The best-performing algorithms are Linear Regression, Random Forest, and Adaboost, with Linear Regression being the leading Algorithm with minimum errors. The worst-performing Algorithms are SVM, Neural Network, and Constant, with constant being the Algorithm with the most errors. We can conclude that, in this case, the best Algorithm is Linear Regression, and the worst Algorithm is SVM.

Table XI
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	MAPE	R2
Linear Regression	0.000	1.000
Gradient Boosting	0.004	1.000
AdaBoost	0.003	0.999
Random Forest	0.003	0.999
Tree	0.005	0.998
KNN	0.260	0.326
Neural Network	0.389	-0.003
Constant	0.367	-0.011
SVM	0.933	-9.982

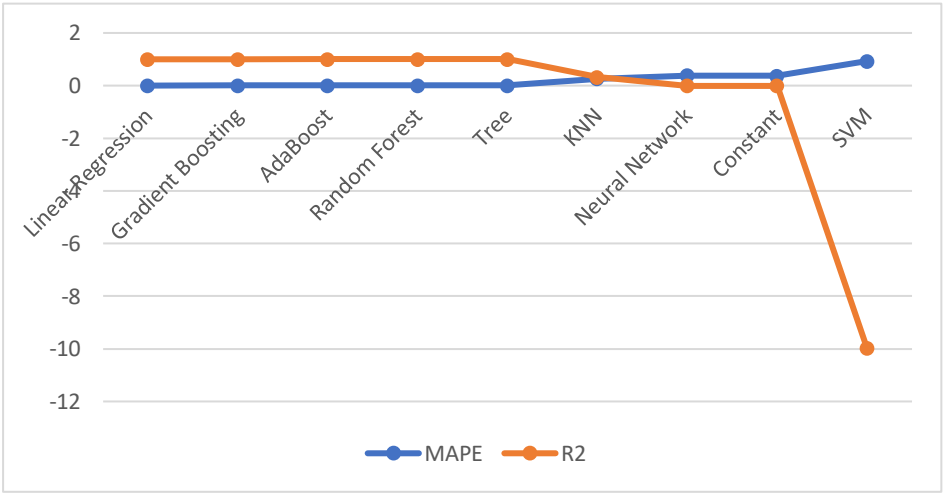


Figure 15. the performance results for the First Dataset p using an 80/20 data split

The following table and graph for the fourth Dataset represent the testing and scoring results of the Dataset on cross-validation with the number of folds 10. The number of errors for each Algorithm varies from 0.015 at the minimum to 25.652 at the maximum. The best-performing algorithms are Linear Regression, Random Forest, and Adaboost, with Linear Regression being the leading Algorithm with minimum errors. The worst-performing Algorithms are SVM, Neural Network, and Constant, with SVM being the Algorithm with the most errors. We can conclude that, in this case, the best Algorithm is Linear Regression, and the worst Algorithm is SVM.

Table XII
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	MAPE	R2
Random Forest	0.015	0.999
Ada Boost	0.016	0.999
Tree	0.018	0.999
Gradient Boosting	0.030	0.999
kNN	0.429	0.590
Neural Network	10.814	-0.001
constant	11.054	-0.001
SVM	25.652	-0.838

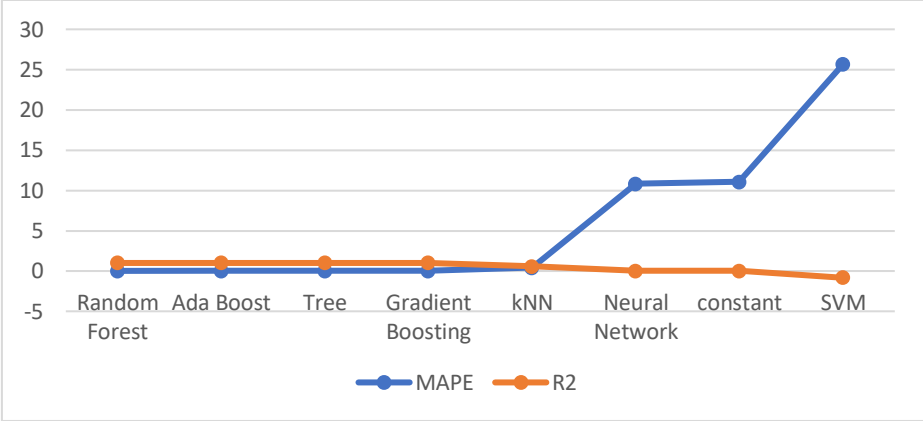


Figure 16. Fourth Dataset performance chart with ten k-fold

The following table and graph represent the testing and scoring results of the Dataset on the random sample with training of 80% and testing of 20%. The number of errors for each Algorithm varies from 0.015 at the minimum to 24.792 at the maximum. The best-performing algorithms are Random Forest, AdaBoost, and Tree, with Random Forest being the Algorithm with the least errors. The worst-performing Algorithms are SVM, Neural Network, and Constant, with SVM being the Algorithm with the most errors. One can conclude that the best Algorithm in this case is Random Forest, and the worst remains SVM.

Table XIII
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	MAPE	R2
Gradient Boosting	0.030	0.999
Random Forest	0.015	0.999
AdaBoost	0.016	0.999
Tree	0.018	0.999
kNN	0.443	0.590
Neural Network	11.320	0.048
Constant	11.303	-0.001
SVM	24.792	-0.712

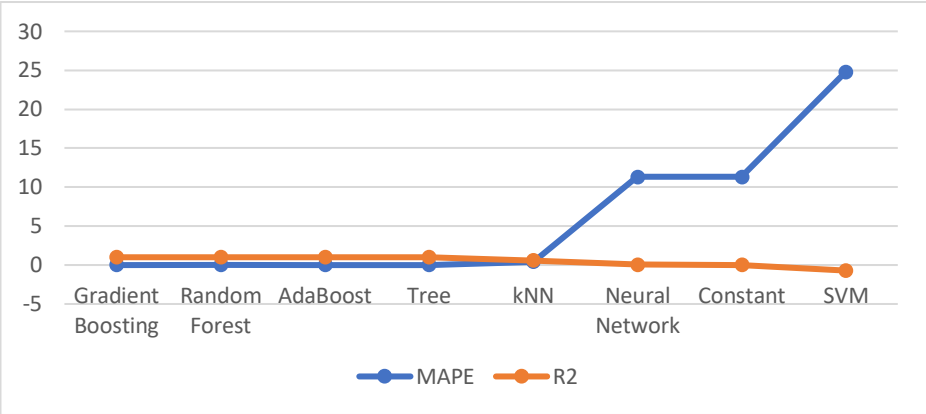


Figure 17. Fourth Dataset performance chart with 80/20 data split

The following table and graph for the fifth Dataset represent the testing and scoring results of the Dataset on cross-validation with the number of folds 10. The number of errors for each Algorithm varies from 0.014 being the minimum to 22.217 being the maximum. The best-performing algorithms are Linear Regression, Random Forest, and Adaboost, with Linear Regression being the leading Algorithm with minimum errors. The worst-performing Algorithm is SVM, which has the most errors. The Linear Regression algorithm is the best model, and SVM is the worst performance-wise.

Table XIV
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	MAPE	R2
Linear Regression	0.014	0.999
Random Forest	0.015	0.999
AdaBoost	0.016	0.999
Tree	0.019	0.998
KNN	0.024	0.998
Gradient Boosting	0.028	0.999
Constant	7.450	-0.001
Neural Network	9.178	0.009
SVM	22.217	-1.211

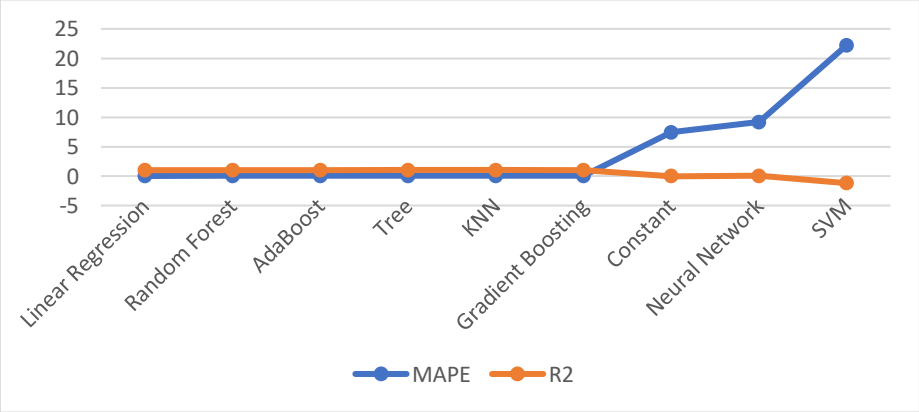


Figure 18. Fifth Dataset performance chart with ten k-fold

The following table and graph represent the Dataset testing and scoring results on random samples with training of 80% and testing of 20%. The number of errors for each Algorithm varies from 0.014 being the minimum to 20.328 being the maximum. The best-performing algorithms are Linear Regression, Random Forest, and Adaboost, with Linear Regression being the leading Algorithm with minimum errors. The worst-

performing Algorithms are SVM and Neural Networks, with SVM being the Algorithm with the most errors. We can conclude that, in this case, the best Algorithm is Linear Regression, and the worst Algorithm is SVM.

Table XV
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	MAPE	R2
Linear Regression	0.014	0.999
Random Forest	0.015	0.999
AdaBoost	0.017	0.999
Tree	0.019	0.998
KNN	0.026	0.998
Gradient Boosting	0.027	0.999
Constant	7.540	-0.001
Neural Network	10.986	-0.813
SVM	20.328	-0.999

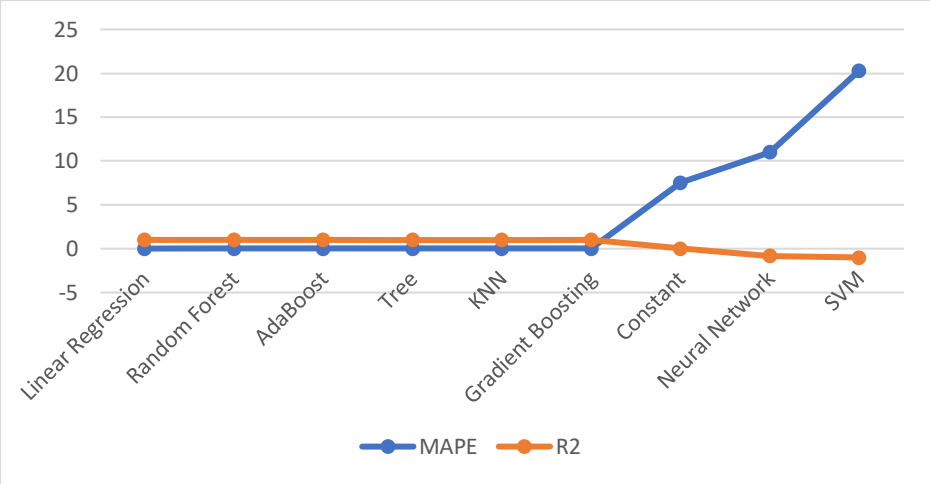


Figure 19. Fifth Dataset performance chart with 80/20 data split

6. Conclusion

Machine learning offers a powerful lens for analyzing the complexities of Bitcoin price prediction. Its algorithms unearth hidden patterns that inform future price trajectories by analyzing historical data. This research explored diverse models like KNN, SVM, Random Forest, and Linear Regression, each contributing unique analytical angles. While Random Forest tackles overfitting and KNN pinpoints data similarities, Linear Regression emerged as the top performer, revealing a remarkably accurate linear relationship between variables for effective price forecasting. This affirms the transformative potential of machine learning for navigating the dynamic world of Bitcoin with greater confidence and informed decision-making.

References

[1] Baltagi, B. H., & Baltagi, B. H. (2011). What Is Econometrics? (pp. 3-12). Springer Berlin Heidelberg.
[2] Gujarati, D. N., & Porter, D. C. (2009). Basic econometrics. McGraw-hill.
[3] Stock, J. H., & Watson, M. W. (2020). Introduction to econometrics. Pearson.
[4] Pintelas, E., Livieris, I. E., Stavroyiannis, S., Kotsilieris, T., & Pintelas, P. (2020). Investigating the problem of cryptocurrency price prediction: a deep learning approach. In Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16 (pp. 99-110). Springer International Publishing.

[5] Poongodi, M., Vijayakumar, V., & Chilamkurti, N. (2020). Bitcoin price prediction using ARIMA model. International Journal of Internet Technology and Secured Transactions, 10(4), 396-406.
[6] Awoke, T., Rout, M., Mohanty, L., & Satapathy, S. C. (2020). Bitcoin price prediction and analysis using deep learning models. In Communication Software and Networks: Proceedings of INDIA 2019 (pp.
[7] El Naqa, I., & Murphy, M. J. (2015). What is machine learning? (pp. 3-11). Springer International Publishing.
[8] Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning. MIT press.
[9] Baştanlar, Y., & Özuysal, M. (2014). Introduction to machine learning. miRNomics: MicroRNA biology and computational analysis, 105-128.
[10] Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. Journal of Computational and Applied Mathematics, 365, 112395.
[11] McNally, S., Roche, J., & Caton, S. (2018, March). Predicting the price of bitcoin using machine learning. In 2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP) (pp. 339-343). IEEE.
[12] Ji, S., Kim, J., & Im, H. (2019). A comparative study of bitcoin price prediction using deep learning. Mathematics, 7(10), 898.
[13] McNally, S., Roche, J., & Caton, S. (2018, March). Predicting the price of bitcoin using machine learning. In 2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP) (pp. 339-343). IEEE.
[14] Kavitha, H., Sinha, U. K., & Jain, S. S. (2020, January). Performance evaluation of machine learning algorithms for bitcoin price prediction. In 2020 Fourth International Conference on Inventive Systems and Control (ICISC) (pp. 110-114). IEEE.
[15] Rane, P. V., & Dhage, S. N. (2019, March). Systematic erudition of bitcoin price prediction using machine learning techniques. In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS) (pp. 594-598). IEEE.
[16] Chen, J. (2023). Analysis of bitcoin price prediction using machine learning. Journal of Risk and Financial Management, 16(1), 51.
[17] Phaladisailoed, T., & Numnonda, T. (2018, July). Machine learning models comparison for bitcoin price prediction. In 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE) (pp. 506-511). IEEE.
[18] Velankar, S., Valecha, S., & Maji, S. (2018, February). Bitcoin price prediction using machine learning. In 2018 20th International Conference on Advanced Communication Technology (ICACT) (pp. 144-147). IEEE.
[19] Maulud, D., & Abdulazeez, A. M. (2020). A review on linear Regression comprehensive in machine learning. Journal of Applied Science and Technology Trends, 1(4), 140-147.
[20] Rigatti, S. J. (2017). Random forest. Journal of Insurance Medicine, 47(1), 31-39.
[21] Ying, C., Qi-Guang, M., Jia-Chen, L., & Lin, G. (2013). Advance and prospects of AdaBoost algorithm. Acta Automatica Sinica, 39(6), 745-758.
[22] Freund, Y., & Mason, L. (1999, June). The alternating decision tree learning algorithm. In icml (Vol. 99, pp. 124-133).
[23] Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. Frontiers in neurorobotics, 7, 21.
[24] Suthaharan, S., & Suthaharan, S. (2016). Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, 207-235.
[25] Sun, S., & Huang, R. (2010, August). An adaptive k-nearest neighbor algorithm. In 2010 seventh international conference on fuzzy systems and knowledge discovery (Vol. 1, pp. 91-94). IEEE.
[26] Ding, S., Su, C., & Yu, J. (2011). An optimizing BP neural network algorithm based on genetic Algorithm. Artificial intelligence review, 36, 153-162.
[27] Binev, P., Cohen, A., Dahmen, W., DeVore, R., Temlyakov, V., & Bartlett, P. (2005). Universal algorithms for learning theory part i: Piecewise constant functions. Journal of Machine Learning Research, 6(9).
[28] Cameron, A. C., & Windmeijer, F. A. (1997). An R-squared measure of goodness of fit for some common nonlinear regression models. Journal of econometrics, 77(2), 329-342.
[29] Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 7, e623.
[30] Tayman, J., & Swanson, D. A. (1999). On the validity of MAPE as a measure of population forecast accuracy. Population Research and Policy Review, 18, 299-322.