Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

Crypto- Currency Price Prediction with Decision Tree Based Regressions Approach

Ali Naghib-Moayed*[1] and Reza Habibi†[2]

[1]Department of Statistics, Allameh Tabatabyee University
[2]Iran Banking Institute, Central Bank of Iran

ABSTRACT

Generally, no one can reject the fact that crypto currency market is expanded rapidly during last few years as, nowadays, crypto currency market is attractive for both traders and business who are not willing to pay for FATF services for transferring money. With this in mind, crypto currency price prediction is crucial for many people and business entities. While there have been quite a few conventional statistical models to forecast crypto currency prices, we decided to make price prediction using decision Tree Based Regression. In this research we devised a decision tree models to predict Bitcoin which is the most renowned and frequently used crypto currency. we used Volume from, Volume to, New addresses, Active addresses, large transaction count, Block height, Hash rate,

*Keyword:*Crypto currency price prediction, Decision Tree, ARIMA.

AMS subject Classification: 62P10

ARTICLE INFO

Article history:
Received 15, January 2020
Received in revised form 17, September 2020
Accepted 19 November 2020
Available online 30, December 2020
Research paper

*ali.ampmk@gmail.com
†Corresponding author: R. Habibi. Email: r_habibi@ibi.ac.ir

1 Abstract continued

Difficulty, Current supply as predictor variables in addition to historical crypto currency price data during the with a total of 1000 Observations. We find that forecasting accuracy of decision tree models are higher than benchmark models such as linear regression and autoregressive integrated moving average(ARIMA).

2 Introduction

Bitcoin is the most valuable and, needless to say, the most important crypto currency in the word. There are lots of traders who trade on Bitcoin in addition to numerous business which transfer their money using Bitcoin block chain to avoid banking system complexities and expenses. The changes in the Bitcion price have a great impact on both traders and businesses that are interacting with Bitcoin. Unfortunately, financial theories and models cannot be used for forecasting the Bitcoin price as no one really have information about the financial procedure which is supporting the Bitcoin. Hence, it's all on statistics to offer a rational method for forecasting the Bitcoin price. Many statistical methods including Regression analysis, Time series analysis and data mining approaches can be used for Bitcoin price prediction. As a popular data mining method, decision tree models have a great predictive power in some studies. Moreover, unlike most of data mining models that are considered as a 'black box', decision tree models are interpretable in theory an application. In this research, we develop some decision tree models to compare with such benchmark models as ARIMA models for forecasting accuracy [1].

2-Decision tree Theory

Being both efficient and robust in addition to enjoying from simple structure are the most attractive decision tree method characteristics. According to J.R. Quinlan (1992), the most precious advantage of decision tree is that it can be easily interpreted after making prediction. Decision tree can be used for both classification and regression analysis. Figure1 depicts the basic decision tree structure.

The Root node on the top of the tree involves the full training dataset. The Leaf nodes are nodes that are located at the end of the tree, while the nodes in between are called intermediate nodes. The root and intermediate nodes will split into subsets based on certain attributes [2]. A decision has to be made whether to split a certain node or leave it as a leaf node. This process continues until the tree is fully grown. If the values of the decision tree leafs are categorical, the produced tree is classification type and if leafs are filed with numeric values the tree is regression tree type. In this article decision trees are going to be used for regression analysis aiming for Bitcoin price prediction
In this section we are going to introduce two different algorithms for running decision tree in addition to going through Random Forest strategy which is a method, designed for decision tree accuracy promotion[3].

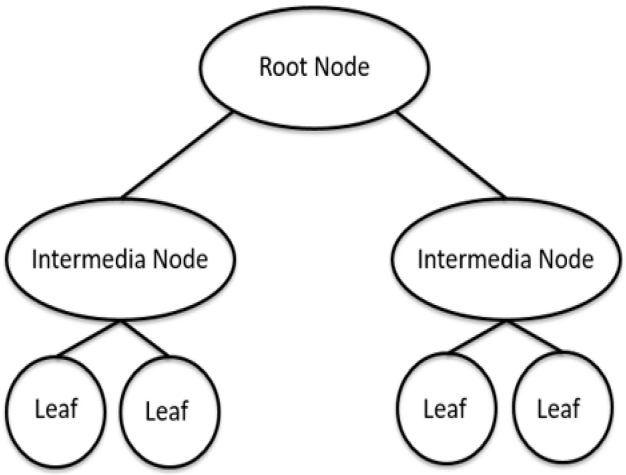

Figure 1: Decision Tree Structure

2.1-CART Regression Tree

The CART Regression Tree model begins with an entire dataset, S and searches every distinct value of each predictor to find the best splitting point which partitions the database into two groups (S1 and S2) such that the overall the sum of squares error (SSE) are minimized

$$SSE = \sum_{i \in S_1} (y_i - \overline{y_1})^2 + \sum_{i \in S_2} (y_i - \overline{y_2})^2$$

Where, $\overline{y_i}$ is the average of training set outcomes within group S_i.

Then, within each groups S1 and S2 this method searches for the predictor and split value that best reduce SSE.

Once the full tree has been grown, the tree may be very large, so it might over fit the training set. The tree is then pruned back to potentially smaller depth. To do this, we simply penalize the error value used as splitting criteria.

$$SSE_{cp} = SSE + cp \times (\#\ Terminal\ Nodes)$$

Where *cp* is equal to complexity parameter while smaller penalties tend to produce more complex models, which in this case, result in larger trees.

To find the best pruned tree we evaluate data across a sequence of *cp* values. This process generates one SSE for each chosen *cp* value. But we know that these SSE values will vary if select different sample of observations. In fact this phenomena which is known as high model variance is the most important drawback of the decision tree process. Using a cross validation strategy is usually helpful in such occasions as we can use it to calculate SSE at each *cp* value, more accurately. Finally, we can choose *cp* that generates the smallest possible RMSE value and use it to promote the regression tree [4].

2.2-M5 Regression Tree

One limitation associated with CART Regression Trees is that the average of observations on each node is introduced as the prediction for all of observations related to that node. Consequently, this models won't offer an appreciable performance in confronting with samples whose true outcomes are extremely high or low. Using different estimator in terminal nodes is a popular approach to dealing with this insufficiency. M5 Regression Tree model is quite similar to CART Regression Tree except:

1. The splitting criteria is different
2. The terminal node use linear model to generate predictions
3. Each prediction is usually a combination of the predictions from different models along the same path through the tree. In this way the model avoids the over fitting problem.

As like as CART Regression Tree, the initial split is found using an extensive over predictors and training set samples, however, unlike these models the expected reduction in the nodes error rate is used instead of overall the sum of squares error (SSE) [5].
Let S denote the entire set of data and let S_1, $S_2 \ldots$. S_p represent the p subsets of the data after splitting, then, the expected reduction in the nodes error rate would be equal to

$$error\ reduction = SD(S) - \sum_{i=1}^{p} \frac{n_i}{n} \times SD(S_i)$$

Where SD is a standard deviation and n_i is number of samples in partition i and for M5 algorithm, p is equivalent to two on each split.
The split that is associated with the largest reduction in error is chosen and a linear model is created within the partitions using the split variable in the model. The error associated with each linear model is used in place of SD(S) in error reduction equation. The tree growing process continues along the branches of the tree until there are no further improvements in the error rate or there are not enough samples to continue the process.
Once the complete set of linear models have been created, each undergoes a simplification procedure to potentially drop some of the terms. For a given model. First, the absolute differences between the observed and predicted data are calculated then multiplied by a term that penalizes models with large numbers of parameters the outcome value is called Adjusted Error Rate.

$$Adjusted\ Error\ Rate = \frac{n^* + p}{n^* + p} \sum_{i=1}^{n^*} \left| y_i - \hat{y}_i \right|$$

Where n^*denotes the number of observations used for building the model and p is equal to number of parameters involved in the model.

Second, each model term is dropped and the adjusted error rate is computed. Terms are dropped permanently from model if the adjusted error rate decreases. This process is independently applied to each linear model.
Model trees also incorporate a type of smoothing to decrease the over fitting potential. During prediction process, the new sample goes down the appropriate path of the tree and moving from bottom up, the linear models along that path are combined. Using the figure 2 as reference, suppose the new sample goes down the path associated with model 3 as well as the linear model in the parent node (model 2 in this case). These two predictions are combined using

$$\hat{y} = \frac{n_{(k)}\widehat{y}_k + C\hat{y}_p}{n_{(k)} + C}$$

Where:
$\hat{y}_k$= prediction of child model (model 3)
$\hat{y}_p$ =prediction of parent model (model 2)
$n_{(k)}$= number of observations in child model (model 3)
C= constant (usually=15)

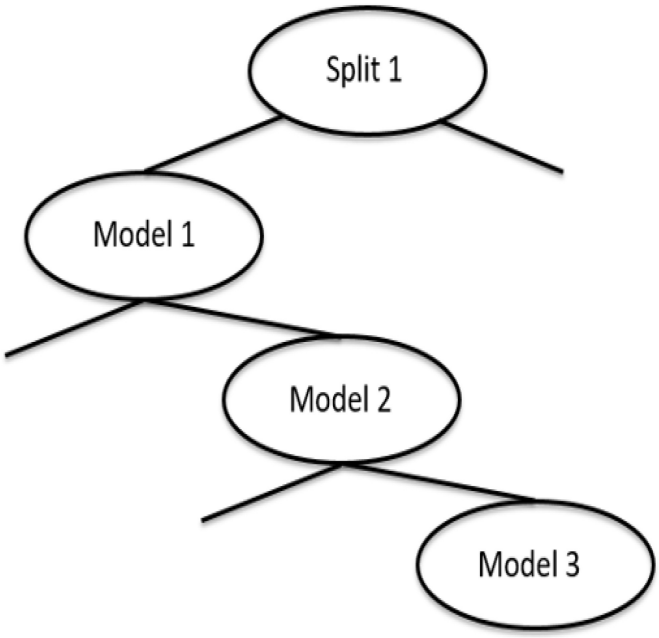

Figure 2: Model

Once this the combined prediction is calculated, it is similarly combined with next model along the tree (in this case, model1) and so on.

2.3-Bagged Tree

Bagging is a general approach that uses bootstrapping in conjunction with any model to construct an ensemble model aiming for reduce the model variance. The method is fairly simple in structure and consists of steps in the following algorithm
Step1: generate a bootstrap sample (generate random sample with replacement for B times)
Step2: generate an unpruned tree model on each bootstrap sample
Step3: calculate the average of B predictions outcome for each new observation and use it as a final prediction.
For models that produce an unstable prediction, like regression trees, aggregating over many versions of data actually reduces the prediction variance and offer more stable predictions.

2.4-Random Forest

Bagging, improves predictive performance over a single tree by reducing variance of the prediction. Generating bootstrap samples introduce a random component into the tree building process, which induces a distribution of trees and therefore also a distribution of predicted values for each sample. The trees in bagging, however, are not completely independent from each other since all of the original predictors are considered at every split of every tree. This characteristic prevents bagging from optimally reducing variance of the predicted values. A random forest method is designed to cover this issue. The method is described in the following algorithm.
Step1: generate a bootstrap sample (generate random sample with replacement for B times)
For i=1 to B do {
Step2: Randomly select k (<p) of the original predictors.
Step3: Generate an unpruned tree model on i'th bootstrap sample.
Step4: Generate and save a prediction based on i'th tree.
}
Step5: calculate the average of B predictions outcome and use it as a final prediction.

3- **Data Collection**

Spanning from 2017 September to 2020 June, We collected 1000 observations of daily data with 9 predictor variables on Bitcoin daily high prices as the target variable. Unfortunately, there is no economic based literature available about independent variables effecting the Bitcoin price, therefore, we used statistical approach for determining suitable pack of independent variables. In fact, we simply used scatter plot between each possible independent variable and dependent variable to identify possible relationship between variables. Variables which depict a noticeable relation with Bitcoin price daily high price will be included in the model at the first step and the model decides to weather keep or omit each of them at the second step. The following pack of plots depicts scatter plot of each probable independent variable against the Bitcoin price daily high variable.
Based on scatter plots following variables chosen to be entered in the model: Volume from, Volume to, New addresses, Active addresses, large transaction count, Block height, Hash rate, Difficulty, Current supply.

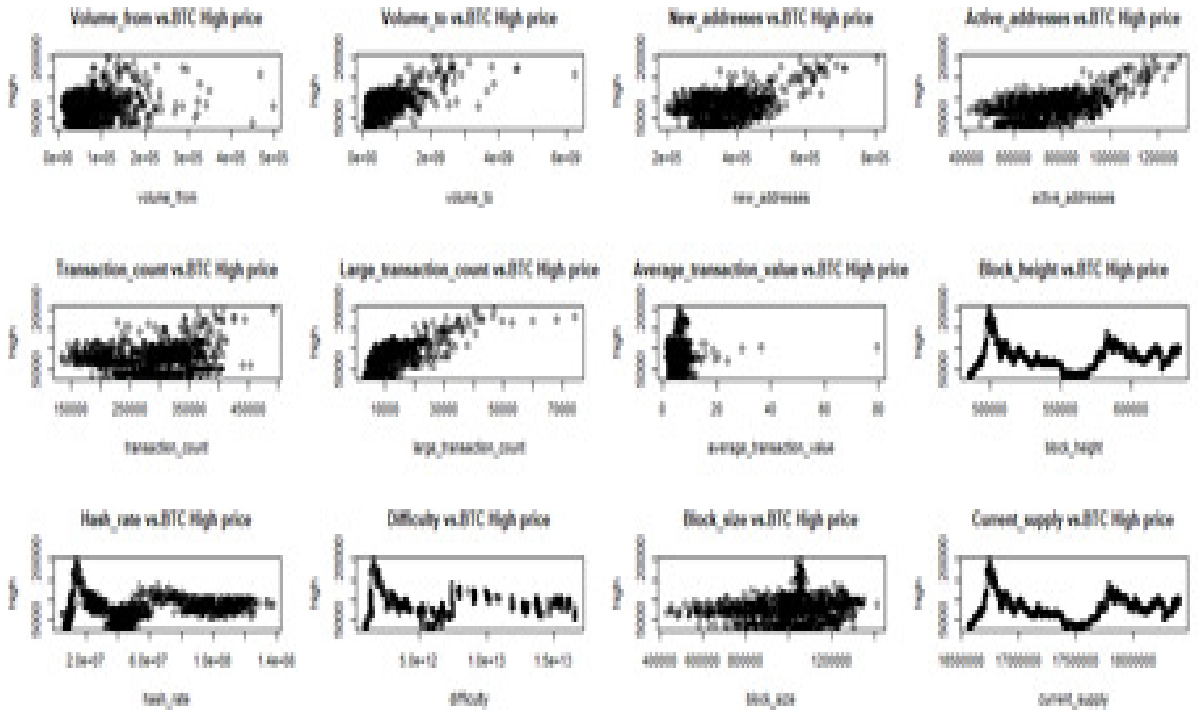

Figure 3:

4-Methodology

With both target and independent variables being continuous, we now deploy CART and M5 regression trees in addition to Random Forest Regression in order to compare their performance in terms of accuracy. ARIMA model is used as benchmark for better comparison. We use R software to run all models and computations in this research.
Based on the data and problem characteristics, usually, there are several criteria's for forecasting error calculation, while, all of them can be interpreted using a single general rule: the lower the prediction error, the better the forecasting accuracy. To compare the performance of each of the models, we use Root Mean Squares Error (RMSE) and Mean Absolute Error (MAE), along with R Square, which measures the explanatory power of the model towards the data. The lower the value of RMSE and MAE, the better the model accuracy, and the higher the R Square, the better explanatory power the model.

5-Analysis and results

Because we are using ARIMA as a benchmark model, we need to both include lagged information in decision tree modeling process in terms of L1 where, L1 stands for one day lagged Bitcoin high price. In addition, we need to include mentioned independent variables into ARIMA model as regression variables in order to be as judicious as possible. Figure4 shows the daily time series plot of the Bitcoin high prices with a total of 1000 observations.

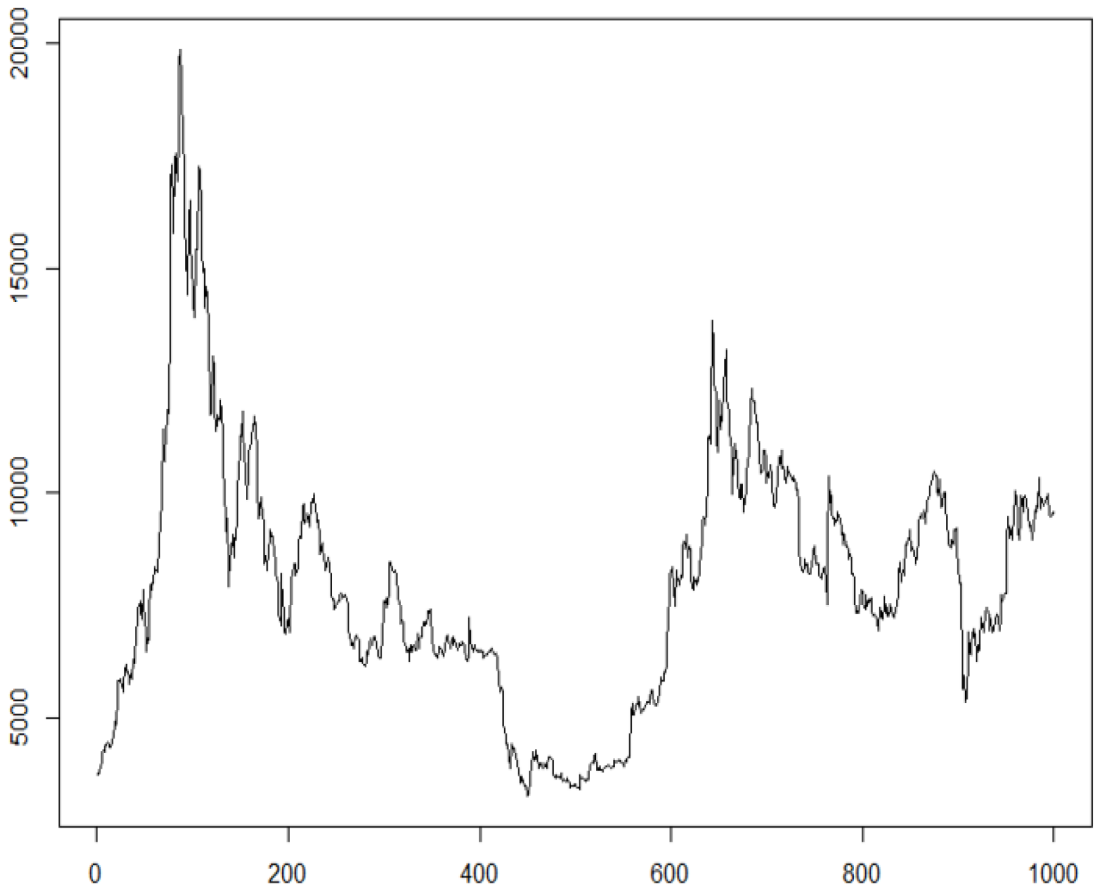

Figure 4: Bitcoin daily high price

In the remaining parts of this section, we are going to have a discussion about the few considerations for running some of mentioned models and at the end, we are going to have comparison between models in terms of accuracy.

5.1-first consideration

Generally, ARIMA is a Time series model and it uses Time series of past observations for forecast generation. Based on Time series definition, being time based ordered is an indispensable characteristic of any Time series. With this in mind, running a K-fold cross validation process for identifying Time series model accuracy, seems to be irrational. In this paper, for fitting ARIMA model and accuracy determination, we use the following algorithm.

1. Consider a value for n which is the window size(we used n=30)
2. Put i=1

For $1 \leq i \leq number\ of\ observations$

3- Run ARIMA for observations which are indexed in [i, i+30] interval.

4- Generate and save prediction for observation number i+31
5- Put i=i+1

1. Calculate accuracy of model by comparing predicted and real values.

5.2-Second consideration
For running the CART Regression Tree model, we used SSE measure introduced in section 2.1 as a splitting criteria. In addition, based on SSE_{cp} formula, we need to determine a suitable value for cp in order to run a pruning process. Generating a series of SSE values by evaluating data across sequence of cp values and choosing a cp which offers minimum SSE is a conventional way for finding a suitable value for cp. The figure shows calculated SSE against different values of cp.

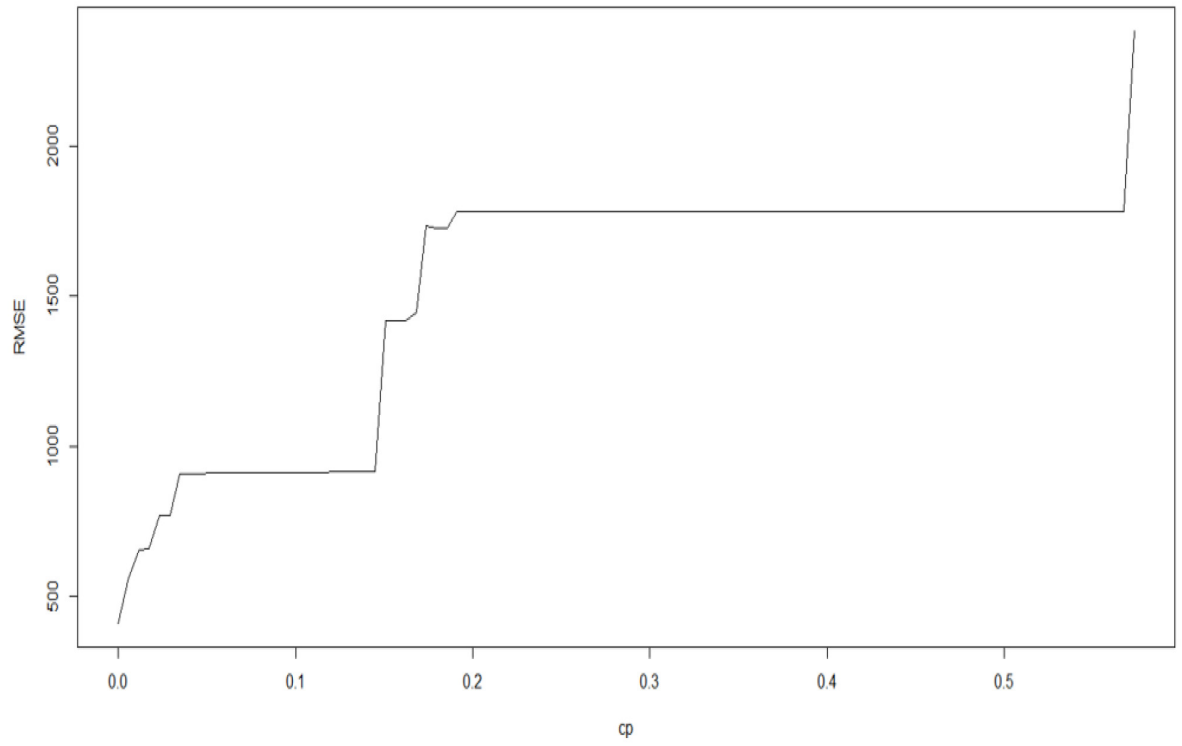

Figure 5: RMSE of CART model against different values of cp

Based on the figure, the best cp value is quite close to zero which means that the final Tree would be bushy. Finally, we used k-fold cross validation for training the model.
5.3-Third consideration
Based on the literature offered on section 2.2, both of pruning and smoothing processes are designed for improving the model accuracy by exporting redundant variables and fighting against over fitting phenomena, respectively. Figure5 depicts the effect of smoothing and pruning on final Tree accuracy.
5.4- Forth consideration

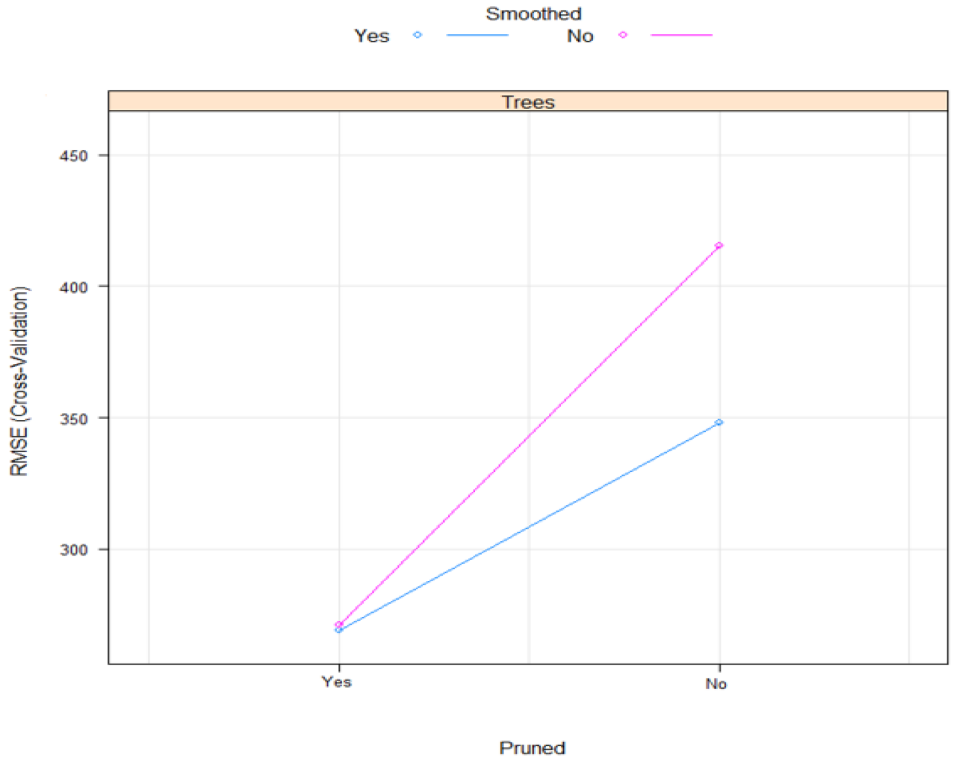

Figure 6: effect of pruning and smoothing on M5 RMSE

Determining the number of variables to be used for building each random tree and the whole number of random trees (population of the forest) are very important, for running Random Forest, in matters of accuracy and time expenses, respectively as it's necessary to run a short investigation to find an appropriate value for mentioned parameters.

Figures 6 and 7 are showing the appropriate values for population of the Random Forest and number of variables that should be used for making each tree of the Random Forest.

Based on figure6, the appropriate population for Random Forest is equivalent to 200 as the accuracy shows no improvement for more populated forests while running more computation time is demanded for running more populated Random Forest. The suitable number of variables for constructing each tree of random forest is 3 as it generates not only the list amount of error but also more simple trees compared with using 4 variables for tree generation.

5.6- Results

We used CART, M5 and Random Forest decision tree models in addition to ARIMA model for Bitcoin daily high price prediction. Following table presents accuracy measures of each model:

Table 1: Results of CART

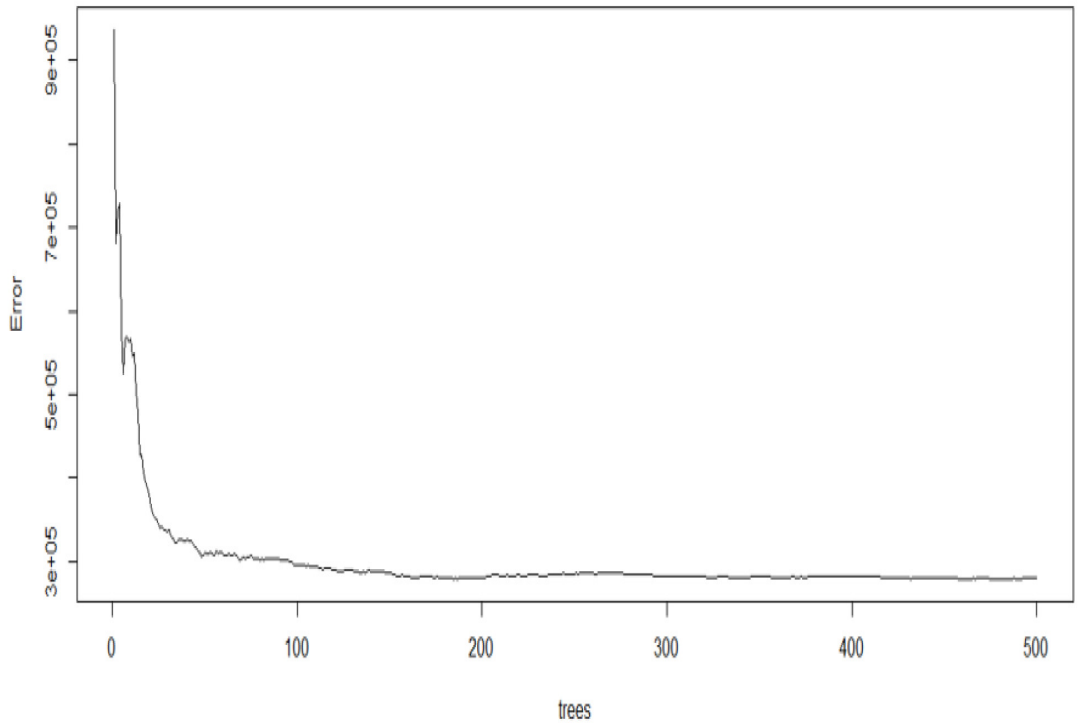

Figure 7: Error values related to Random Forest population

	R^2	RMSE	MAE
CART	.98	412	280.23
M5	.99	268.715	186
Random Forest	.9843	232.62	175
ARIMA	.97	366.60	253

As shown in Table, M5, R_Forest, and CART are decision tree models, whereas ARIMA which is a time series model. It is seen in Table 1 that the original Classification and Regression Tree (CART) model performs the worst among all four models in this research, with R^2 = 0.98, MAE = 280.23, and RMSE = 412, which does worse than the benchmarking ARIMA, although, ARIMA R^2 is not as high as CART R^2. However, the difference is negligible. Since M5 is an improved CART model, it is no surprise that M5 outperforms CART, and even better than the benchmark model of ARIMA, with R^2 =.99, MAE = 186, and RMSE = 268. It is seen from Table that R_Forest is the best model of all the five models in this study with MAE = 175 and RMSE = 232.62 and R^2 = 0.9843.

Figure 5 presents a sample random forest model since it is impossible to show as many as 100 trees as a complete random forest model. The prediction process of the random forest model can be summarized below.

6- Conclusion

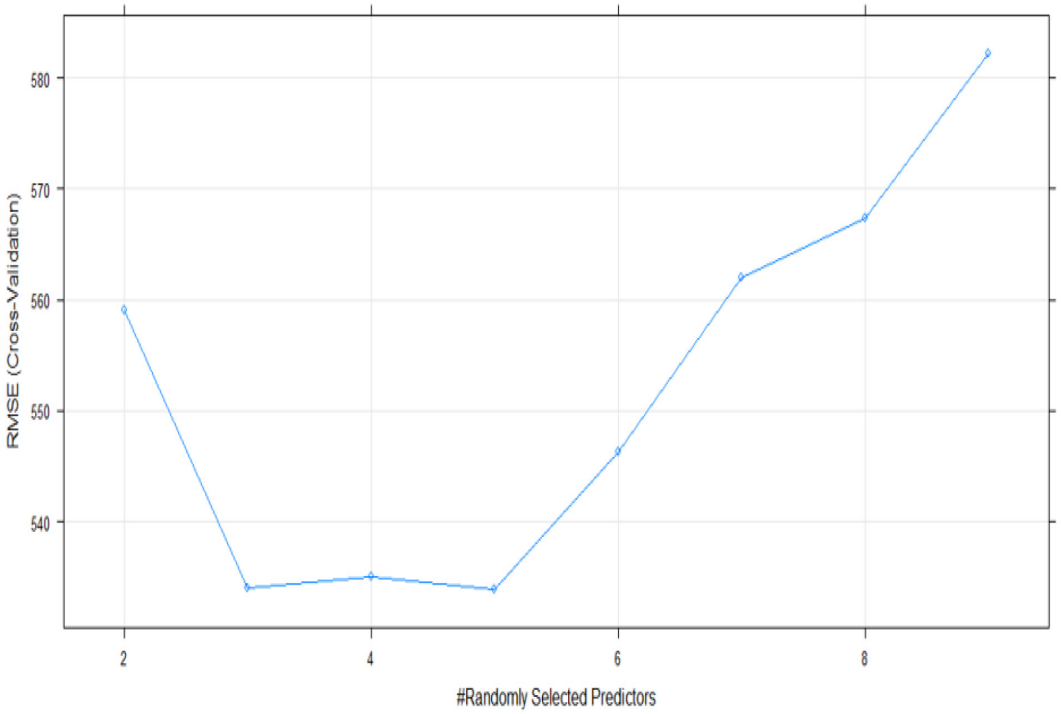

Figure 8: Error values related to Random Forest population

In this research, we compared three decision tree models, Classification and Regression Tree (CART), M5, and Random Forest, with one benchmarking time series autoregressive integrated moving average (ARIMA) model on Bitcoin daily high prices as the target variable during the period 2017 September to 2020 June, with 9 predictor variables. Forecasting accuracy is measured in terms of r squares (R^2), root mean squared error (RMSE), and mean absolute error (MAE).
We found that the Random Forest model is the most superior among all four models studied in this research in terms of forecasting accuracy. It is worth noting that the original classification and regression tree (CART) should not be used for predicting continuous target variables due to its lowest forecasting accuracy among all four models in this research. In fact, the CART model in our research performs much worse than the benchmarking model, ARIMA. As to the M5 model, it is somewhere in the middle, slightly better than the ARIMA model but not as good as the Random Forest model.

References

[1] Diebold FX, Yılmaz K. On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics* 182,

(2014), 119–134.

[2] Kristoufek L. *What are the main drivers of the Bitcoin price*? Evidence from wavelet coherence analysis. (2015). Springer. USA.

[3] Nakamoto S. *Bitcoin: A peer-to-peer electronic cash system*. (2008). Springer. USA.

[4] Narayanan A, Bonneau J, Felten E, Miller A, Goldfeder S. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press; (2016).

[5] Schweitzer F, Fagiolo G, Sornette D, Vega-Redondo F, Vespignani A., White DR. Economic networks: The new challenges. *Science* 325, (2009), 422–425.