



Comparative Performance of Machine Learning Ensemble Algorithms for Forecasting Cryptocurrency Prices

V. Derbentsev^a, V. Babenko^{*b}, K. Khrustalev^c, H. Obruch^d, S. Khrustalova^c

^a Informatics and Systemology Department, Institute Information Technologies in Economics, Kyiv National Economic University named after Vadim Hetman, Kyiv, Ukraine

^b International E-commerce and Hotel & Restaurant Business Department, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine

^c Department of Computer-Integrated Technologies, Automation and Mechatronics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

^d Department of Economics and Management of Industrial and Commercial Business, Ukrainian State University of Railway Transport, Ukraine

PAPER INFO

Paper history:

Received 06 July 2020

Received in revised form 13 August 2020

Accepted 03 September 2020

Keywords:

Cryptocurrencies Time Series

Short-term Forecasting

Machine Learning

Random Forest

Gradient Boosting

ABSTRACT

This paper discusses the problems of short-term forecasting of cryptocurrency time series using a supervised machine learning (ML) approach. For this goal, we applied two of the most powerful ensemble methods including Random Forests (RF) and Stochastic Gradient Boosting Machine (SGBM). As the dataset was collected from daily close prices of three of the most capitalized coins: Bitcoin (BTC), Ethereum (ETH) and Ripple (XRP), and as features we used past price information and technical indicators (moving average). To check the effectiveness of these models we made an out-of-sample forecast for selected time series by using the one step ahead technique. The accuracy rate of the forecasted prices by using RF and GBM were calculated. The results verify the applicability of the ML ensembles approach for the forecasting of cryptocurrency prices. The out of sample accuracy of short-term prediction daily close prices obtained by the SGBM and RF in terms of Mean Absolute Percentage Error (MAPE) for the three most capitalized cryptocurrencies (BTC, ETH, and XRP) were within 0.92-2.61 %.

doi: 10.5829/ije.2021.34.01a.16

1. INTRODUCTION¹

Cryptocurrencies are one of the popular modern financial assets. Despite the fact that in the last decade, since the appearance of the first cryptocurrency namely Bitcoin, their exchange rates and market capitalization have undergone several dramatic ups and downs. The total market capitalization of cryptocurrencies amounted to \$ 15.6 billion at the beginning of 2017; at the beginning of 2020 it was about \$ 230 billion, and the maximum market cap had reached almost a trillion dollars (\$ 860 billion) in the middle of 2018 [1].

The role and position of cryptocurrencies in the global financial market is a controversial and debatable

topic [2-4]. Significant fluctuations in their prices and legal uncertainty of the transactions performed with them in most countries causes significant uncertainty and, as a result, high investment risk of these assets.

A vast majority of economists and researchers inclined to believe that cryptocurrencies are speculative financial assets, intended primarily for short-term investments horizon (see, for example [2-6]). That's why for decision-making in the cryptocurrency market it would be necessary to develop adequate forecasting tools.

It should be noted, that cryptocurrency time series are characterized by high volatility, non-Gaussian distributions, heavy tails and the presence of abnormal and extreme events [7-8]. At the same time, the key drivers which determine the price of cryptocurrencies are still poorly understood and identified [4-6].

*Corresponding Author Institutional Email:
vitalinababenko@karazin.ua (V. Babenko)

Therefore classical approaches based on statistical frameworks, time-series and econometric models, which were efficiently used for modeling and forecasting traditional assets (fiat currencies, commodity prices, securities value, etc.), in the case of cryptocurrencies have proven to be ineffective.

Thus, the main purpose of our research is exploring and comparing the efficiency of Machine Learning (ML) ensembles-based approaches, such as Random Forest (RF) and Gradient Boosting Machine (GBM) on the problem of short-term forecasting of cryptocurrency prices.

2. LITERATURE REVIEW

Recently, ML methods and algorithms, which have shown significant effectiveness in many areas of complex systems research (biomedicine, neurorobotics, image and voice analysis, pattern recognition, machine translation, etc.) [9-12], have been applied to financial time series analysis [13-16].

The most common ML algorithms in financial forecasting are Artificial Neural Networks (ANNs) of various architectures [16-20], and Support Vector Machines (SVM) [20-23].

These approaches have proven to be effective for the forecasting of financial assets [19-24] and cryptocurrencies [18, 23, 25-28].

Several studies [18, 26, 28] presented the results of predicting cryptocurrency exchange rates by using ARIMA (as baseline) and different ML approaches. Summarizing their results allows us to conclude that ML algorithms outperform time series models in predicting both cryptocurrency prices (or returns) and their volatility.

There is a number of research papers (see, for example, Makridakis et al., [13]; Bontempi et al., [14]; Persio and Honchar, [15]), which presented results stating that ANN outperformed other ML methods at forecasting cryptocurrencies prices.

However, Hitam and Ismail [27] compared forecasting performance of different ML algorithms by using cryptocurrency time series (prices). They tested ANNs, SVM and Boosted NN for top-six cryptocurrencies and conducted that SVM has better predictive accuracy (in terms of Mean Absolute Percentage Error, MAPE) than others.

Mallqui and Fernandes [28] examined Bitcoin price direction and daily exchange rate predictions using ANN and SVM. Regarding obtained results, the SVM presented the best performance for both price trend change (classification problem, accuracy 59.45%) and exchange rate predictions (regression problem, MAPE within 1.52-1.58%).

During the last few years the main attention of researchers has been focused on the application of Deep

Learning (DL) approaches in the field of financial forecasting, including cryptocurrencies [18, 29-33].

Sezer et al. [29] recently prepared a detailed overview devoted to applying DL techniques for forecasting financial time series.

McNally [18] focused on predicting Bitcoin trends (classification problem) by using Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM). He used only past prices and several technical indicators such as the simple moving average for prediction. The accuracy that he obtained for predicting changes in the direction of the Bitcoin trend was within 51-52% for both of the methods.

Kumar and Rath [22] compared the forecasting ability of LSTM and Multi-Layer perceptron (MLP) for predicting the direction of price changes in Ethereum. They used daily, hourly, and minute data and concluded that LSTM requires significantly more training time; despite this it did not significantly outperform MLP.

In the study of Yao et al. [31], devoted DL forecasting cryptocurrency prices, was used extended dataset, which included lagged prices value, market cap, trading volume, circulating and maximum supply. According to their results the prediction accuracy has been significantly increased when a wider dataset is used.

Chen et al. [32] developed two stage forecasting strategy. On the first stage they used ANN and RF for feature selection of several economic and technological factors, and on the second stage they made predictions on the Bitcoin exchange rate by using LSTM. As to their results, LSTM had better forecasting performance than ARIMA and SVM. Moreover, incorporating economic and technological factors as additional predictors increases prediction accuracy.

It should be noted that if we are solving the complex problem of regression (forecasting) or classification, it often happens that none of the models provides the desired quality and accuracy. In this cases, we can build an ensemble of individual models (algorithms) in which the errors of each other are mutually compensated.

This idea is based on another powerful class of ML approaches of designing ensembles C&RT: Random Forest (RF) [34-35] and Gradient Boosting Machine (GBM) [36-37], which used bagging (RF) and boosting (GBM) technique. Both RF and GBM are powerful methods that can efficiently capture complex nonlinear patterns in data.

But much less attention has been paid to these algorithms in the field of financial time series analysis. Thus, Varghade and Patel [21] tested RF and SVM to forecast stock market index S&P CNX NIFTY. They noted that the Decision Trees model outperforms the SVR, although RF at times is found to overfit the data.

Kumar and Thenmozhi [22] explored set of classification models for predicting direction of index

S&P CNX NIFTY. Their empirical results suggest that both the SVM and RF outperforms the other classification methods (NN, Linear Discriminant Analysis, Logit), in terms of predicting the direction of the stock market movement, but at the same time SVM it turned out to be more accurate.

Recently appeared several papers devoted to applying ensembles approaches for forecasting cryptocurrency prices [38-40]. Borges and Neves [38] tested four ML algorithms for price trend predicting: LR, RF, SVM and GBM. All learning algorithms outperform the Buy and Hold investment strategy in cryptomarket. The best result was obtained by ensembles voting (accuracy 59.3%).

Chenet et al. [39] applied a set of learning models including RF, XGBoost, Quadratic Discriminant Analysis, SVM and LSTM for Bitcoin 5-minute interval and daily prices. The authors used a large dataset including technological, market and trading, socio-media and fundamental factors. A somewhat unexpected result was that for daily prices, better results were obtained by using statistical methods (average accuracy 65%) as compared to ML methods (average accuracy 55.3%). Among ML models the SVM was the best performing, with an accuracy of 65.3%.

Sun et al. [40] developed a novel method of the price prediction trend of cryptocurrency market (42 coins) by using modification of GBM (LightGBM). The authors, besides past prices, also included several macroeconomic factors: Dow Jones index, S&P 500 index, WTI crude oil price index and others which affect the price fluctuation of cryptocurrency market. They tested different time period (2-days, 2-weeks, 2-month) and the best performance has been received for two week forecasting time horizon: accuracy whitening 0.52 (RF) to 0.61 (SVM, LightGBM).

For instance of rewired papers [38-40], which examined the prediction of price trend changes (classification problem), we focused on forecasting exchange rates (prices) by using RF and GBM (regression problem). Moreover, we used the stochastic GBM (SGBM), which has some advantages such as less learning time, less used memory and higher accuracy.

3. METHODOLOGY

3. 1. Supervised Machine Learning Approach

In our study we will apply supervised machine learning technique to forecast cryptocurrency time series. Consider a sample of pairs of features $\mathbf{x} = (x_1, x_2, \dots, x_p, \dots, x_M)$ (lagged daily prices $y_{i-1}, y_{i-2}, \dots, y_{i-p}$, $i > p$ and several technical indicators x_{p+1}, \dots, x_M), and the target variable (price on the next day) $y : (\mathbf{x}_i, y_i)_{i=1,2,\dots,n}$ length n .

We believe that between the target and features there is an unknown functional dependence f , which can be parameterized by an approximation function:

$$y \approx \hat{f}(\mathbf{x}) = f(\mathbf{x}, \theta). \quad (1)$$

Let $L(y, f)$ be the loss function, which characterizes the deviations between the actual and predicted values of target variable. Thus, our task is to minimize the loss function in the sense of mathematical expectation on the available dataset:

$$\begin{aligned} \hat{f}(\mathbf{x}) &= \arg \min_{f(\mathbf{x})} L(y, \hat{f}(\mathbf{x})) = \\ &= \arg \min_{f(\mathbf{x})} E_{xy} L(y, f(\mathbf{x}, \theta)) = f(\mathbf{x}, \theta) \end{aligned} \quad (2)$$

For solving (2) we have to optimize $L(\cdot)$ using parameters θ

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} E_{xy} L(y, f(\mathbf{x}, \theta)) = \\ &= E_{xy} [E_y (L(y, f(\mathbf{x}, \theta)) | \mathbf{x})]. \end{aligned} \quad (3)$$

In this context we can approximate the empirical loss function for N steps using:

$$L_{\theta}(\hat{\theta}) = \sum_{i=1}^N L(y_i, f(\mathbf{x}_i, \hat{\theta})), \quad (4)$$

$$\hat{\theta} = \sum_{i=1}^N \hat{\theta}_i. \quad (5)$$

There are a lot of numerical technics to minimize loss function $L_{\theta}(\hat{\theta})$. The efficient approach is the gradient descent method, according to which we need to calculate gradient of loss function on each step $t = 1, 2, \dots, N$:

$$\nabla L_{\theta}(\theta) = \left[\frac{\partial L(y, f(\mathbf{x}, \theta))}{\partial \theta} \right]_{\theta=\hat{\theta}}. \quad (6)$$

It should be mentioned that when we employ an ML method it is necessary to solve the problem of Bias-Variance trade-off.

This is the problem of simultaneously minimizing two sources of error that prevent supervised learning algorithms from generalizing beyond their training set [9]:

- The *bias* is an error from erroneous assumptions in the learning algorithm, high bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting);
- The *variance* is an error from sensitivity to small fluctuations in the training set, high variance can cause overfitting, i.e., modeling the random noise in the training data, rather than the intended outputs.

Therefore, when adjusting the model parameters, we have to find a compromise between the forecast error caused by its *bias* and the unstable parameter values (high variance).

Consider the quadratic loss function:

$$L(y, f(\mathbf{x}, \theta)) = \frac{1}{2} \sum_{i=1}^l (y_i - f(\mathbf{x}_i, \theta))^2. \quad (7)$$

Then the mean square forecast error *PE* can be represented in the form of sum of three terms:

$$PE = E_{xy} \left[(y_i - f(\mathbf{x}_i, \theta))^2 \right] = \text{Bias}^2(\hat{f}) + \text{Var}(\hat{f}) + \sigma^2 \quad (8)$$

The first component characterizes the bias of the training method, that is, the deviation of the average response of the trained algorithm from the response of the ideal algorithm. The second component characterizes variance, i.e., the scatter of the responses of trained algorithms compared to the average response.

The third component characterizes the noise in the data and is equal to the error of the ideal algorithm; therefore, it is impossible to construct an algorithm having a lower standard error.

Our main goal is to make a one-step-ahead forecast cryptocurrencies price based on available data set, their past values and compare predictive ability of different ML algorithms for solving this task. Thereby we investigate the ML methods including Random Forest (RF) and Gradient Boosting Machine (GBM). These ensemble methods based on using a set of “weak learning” algorithms, in general Decision Trees (C&RT). Despite the fact that each of them has low accuracy (both for classification and regression), we can get enough strength by combining the base weak learners into ensembles model, which allows to achieve a better forecasting performance. This is because an ensemble technique helps reduce bias and / or variance described above (8).

3. 2. Random Forest

RF proposed by Leo Breiman and colleagues [34, 35] is one of the most powerful ML approach. It is based on bagging technique that uses compositions of basic algorithms. Training data set is divided into many random subsets with replacement (bootstrap samples) and each of base classifier is trained on its own sub-set. The final classifier $a_N(\mathbf{x}, \theta)$ is built as the average of the basic algorithms $h_i(\mathbf{x})$ (for regression):

$$a_N(\mathbf{x}, \theta) = \frac{1}{N} \sum_{i=1}^N h_i(\mathbf{x}, \theta), \quad (9)$$

where N represents the number observations (samples) in training set $(\mathbf{x}_i, y_i)_{i=1,2,\dots,N}$.

Unlike boosting, which will be described in the next section, all elementary classifiers used in bagging are

trained independently. The idea is that the classifiers do not correct each other's mistakes, but compensate them by voting. Basic classifiers must be independent (uncorrelated), so they can be classifiers based on different groups of methods (for example, linear regression, decision trees, neural networks and so on) or trained on independent data sets. In the second case, we can use the same basis algorithm.

Thus bagging efficiency is achieved by training the basic algorithms in different sub-sets obtained by bootstrap. Also in RF the feature used for branching in the certain node is not selected from all possible features set, but only from their random subset of size m . Moreover they are selected randomly, typically with replacement, that's why the same feature can appear several times, even in one branch. The number of features for training each tree is recommended to choose as $m \approx \frac{M}{3}$ (for regression task), where M is the total number of features.

The base models alone are not very accurate, but their ensemble allows significantly improves the quality of the prediction.

RF uses fully grown decision trees. It tackles the error reduction task by reducing variance. The trees are made uncorrelated to maximize the reduction in variance, but the algorithm cannot reduce bias (which is slightly higher than the bias of an individual tree in the forest).

It should be noted that RF are used in deep trees because basic algorithms require low bias; the spread is eliminated by averaging the responses of various trees.

3. 3. Gradient Boosting Machine

Over the past two decades boosting has remained one of the most popular ML methods along with NN and SVM. Boosting, in contrast to bagging, does not use simple voting but a weighted one. The major attractions of boosting are that it is easy to design computationally efficient weak learners. A very popular type of weak learner is a shallow decision tree: a decision tree with a small depth limit.

In following, Friedman [36, 37] will cover the basic steps of GBM. By analogy with RF (see expression (9)) we will build a weighted sum of N basic algorithms

$$a_N(\mathbf{x}) = \frac{1}{N} \gamma_N \sum_{i=0}^N h_i(\mathbf{x}, \theta) \quad (10)$$

and the initial algorithm $h_0(\mathbf{x}, \theta)$ can be defined, for example, in the following form:

$$h_0(\mathbf{x}, \theta) = \frac{1}{l} \sum_{i=1}^l y_i, \quad (11)$$

where l is the number of samples in training set ($l < n$, as a rule $l \approx 0.7 \div 0.8n$).

Assume we have already built ensemble $a_{N-1}(\mathbf{x}, \theta)$ of $N-1$ classifiers on the $N-1$ step. Then we select the next basic algorithm $h_N(\mathbf{x})$ in such a way that the error given by (13) can be reduced as much as possible:

$$\sum_{i=1}^l [L(y_i, a_{N-1}(\mathbf{x}_i, \theta) + \gamma_N h_N(\mathbf{x}_i, \theta))] \rightarrow \min_{\gamma_N, h_N} \quad (12)$$

If we choose $h_N(\mathbf{x}_i, \theta)$ in the following form

$$h_N(\mathbf{x}, \theta) = \arg \min_{h(\mathbf{x}, \theta)} \sum_{i=1}^l (h(x_i, \theta) - s_i)^2, \quad (13)$$

and put the deviation s_i equal to the anti-gradient of the loss function $L(\cdot)$

$$s_i = - \left. \frac{\partial L(\cdot)}{\partial z} \right|_{z=a_{N-1}(x_i)} \quad (14)$$

we will take one step of the gradient descent, which is being implemented by moving towards the fastest reduction of the loss function. Thus, we perform gradient descent in the 1-dimensional space of algorithm predictions on the objects of the training set.

As soon as a new basic algorithm is found, it is possible to select its coefficient γ_N by analogy with the gradient descent:

$$\gamma_N = \arg \min_{\gamma} \sum_{i=1}^l L(y_i, a_{N-1}(\mathbf{x}_i, \theta) + \gamma h_N(\mathbf{x}_i, \theta)) \quad (15)$$

Note that unlike the RF, GBM is prone to retraining, which leads to increase errors on the test sample and on out of sample data. Because of this, as a rule, shallow decision trees are used in boosting. These trees have a large bias, but are not inclined to overfitting.

One more of the effective ways to solve this problem is to reduce the step: instead of moving to the optimal point in the direction of the anti-gradient, a shortened step is taken by

$$a_N(\mathbf{x}, \theta) = a_{N-1}(\mathbf{x}, \theta) + \lambda(\gamma_N h_N(\mathbf{x})), \quad (16)$$

where $\lambda \in [0,1]$ is the learning rate.

It should be noted that both of the above described ensembles approaches (RF and GBM) have their own advantages and disadvantages. Boosting works better on large training samples. It can properly reproduce the boundaries of classes with complex shape. Bagging is preferable for short training sets, it also allows efficient parallelization of calculations, while boosting is performed strictly sequentially.

GBM is based on weak learners (high bias, low variance). In terms of decision trees, weak learners are shallow trees, sometimes even as small as decision stumps (trees with two leaves).

The main advantage of boosting is that it reduces both variance and bias in forecasting, nonetheless the

reduction of the forecast error is still carried out mainly due to the reduction in bias. That's why the bias correction leads to a greater risk of overfitting. It can be argued that in financial applications, RF based on bagging is usually preferable to boosting. Bagging solves the overfitting problem, while boosting solves the underfitting one.

4. DATASET

To limit our analysis to the most popular cryptocurrencies, we use the daily close prices and trading volumes of the three most capitalized coins: Bitcoin (BTC), Ethereum (ETH) and Ripple (XRP). Our initial dataset covers the period from 01/01/2015 to 31/12/2019 for BTC and XRP (1826 observations), and from 07/08/2015 to 31/12/2019 for ETH (1608 observations) according to the Yahoo Finance [41]. Figure 1 shows the daily closing prices for the selected coins.

For training the models, fitting and tuning their parameters, the dataset was divided into training and test subsets in ratio of 80 and 20%. Moreover, the last 92 observations (from 01/10/2019 to 31/12/2019) were reserved for validation which was performed by out-of-sample one-step ahead forecast BTC, ETH, and XRP prices.

5. EMPIRICAL RESULTS

5. 1. Feature Selection Since we focus on ML approach of forecasting cryptocurrencies data, the main purpose of our paper is to get the most accurate one-step ahead forecast of daily prices, based on their past values and several other factors.

According to some empirical studies devoted forecasting cryptocurrencies prices [9, 42, 43], there is a

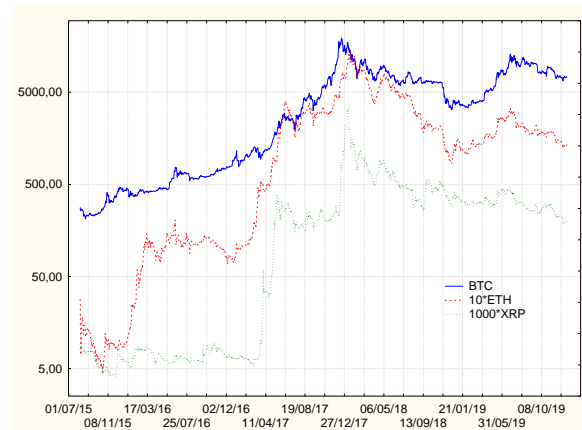


Figure 1. Daily closing prices for BTH, ETH, and XRP (\$ USD) from 07/08/15 to 31/12/19 (log-scale)

seasonal lag which is a multiple of 7 if we use daily observations. In our opinion, this is due to the fact that cryptocurrencies (unlike traditional financial assets, for which a lag of 5 is often observed) are traded 24/7. The results of our correlation analysis also showed the existence of statistically significant lags, multiples of 7, for selected coins.

Therefore, we used the previous values of daily prices for the last four weeks as predictors (lagged daily prices $y_{i-1}, y_{i-2}, \dots, y_{i-28}$). To take into account the changing trends, we used moving average prices of different orders: “fast” – 3,5,7 days, and “slow” – 21, 28 and 35 days.

In addition, we also included in the dataset two exogenous variables: daily trading volumes and the growth rates of daily trading volume with a lag of one day. Thus, the final dataset contains 36 features.

5. 2. Hyper-parameters Tuning It should be noted that hyper-parameters tuning is an important and sophisticated step of the model design. First of all, it is necessary to choose the functional form of the loss function given by Equation (5). To cover the main purpose of our study, the quadratic loss (8) was selected, which generally used for solving the regression problem.

For both of the methods (RF, and GBM), the data is randomly partitioned into training and testing sets. A GBM modification that uses such a partition is called Stochastic GBM (SGBM) [37], which we applied in this study. The training sample is used to fit the models by adding simple trees to ensembles. Testing set is used to validate their performance. For regression tasks, validation is usually measured as the average error. We selected 30% of the dataset as test cases for both the approaches.

Since the RF is not inclined to overfitting, one can choose a large number of trees for the ensemble. We designed RF model with 500 trees. At the same time, in order for the model to be able to describe complex nonlinear patterns in data, it is necessary to use complex trees. So we have chosen a maximum of 15 levels.

Other important parameter for RF is the number of features to consider at each split. As noted in Section 3.2 for regression task it is recommended to choose this value as $m \approx \frac{M}{3}$, where M is the total number of features. We tested different RF models with value m between 8 and 12.

As a stop condition for the number of trees in SGBM (boosting steps) we took the number of trees at which the error on the test stops decreasing. This is necessary in order to avoid the overfitting. For boosting, unlike the RF, the simple trees are usually used. That's why we fitted maximum number of levels in trees and

number of terminal nodes by the criteria of lowest average squared error on both training and test samples.

The final values of hyper-parameters setting are reported in Table 1.

An important parameter for GBM is the learning rate (shrinkage). Regularization by shrinkage consists of modifying the update rule (16) by tuning λ . We selected this value on the grid search according to minimum prediction error on the test set.

The natural logarithm is derived from all features for stabilizing the variance. This is special case of Box-Cox transform.

5. 3. Forecasting Performance The short-term forecasts were made for the selected coins using the absolute values of prices. The target variable is the prediction value of close prices for each cryptocurrencies in the next time period (day) although we used daily observation. Both SGBM and RF models were trained with the same set of features.

For testing efficiency of both approaches, we carried out prediction prices of the selected coins on the hold out last 91 observations by using one-step ahead forecasting technique. The final results are shown in Figures 2-4.

Analysis of the graphs allows us to conclude that both of the ensemble approaches generally well approximate cryptocurrencies time series dynamics, but one can see a certain delay in the model graphs in comparison to the real data.

For comparing prediction performance of different ensembles (SGBM, and RF) we applied Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). Table 2 shows summary of the estimation accuracy for our models using these metrics.

Thus, we can conclude that both methods have the same order of accuracy for the out-of-sample dataset prediction, although boosting is somewhat more

TABLE 1. Final hyper-parameters setting

Parameters	RF	GBM
Loss-function	quadratic	quadratic
Training / test subsamples proportion, %	70/30	70/30
Random subsample rate	0.7	0.7
Number of trees in ensemble	500	250
Maximum number of levels in trees	15	4
Maximum number of features to consider at each split	12	-
Maximum number of terminal nodes in trees	150	15
Minimum samples in child nodes	5	-
Learning rate (shrinkage)		0.1

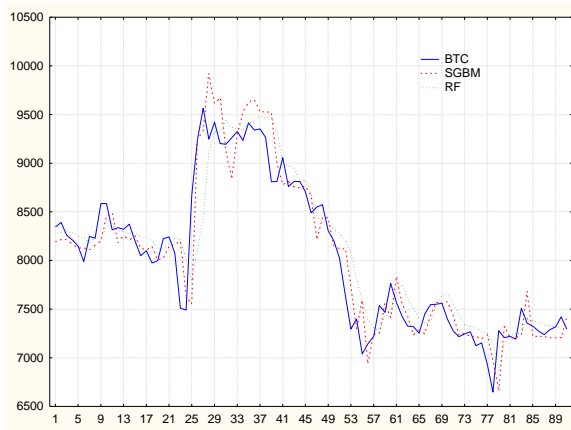


Figure 2. Out of sample prediction of BTC prices

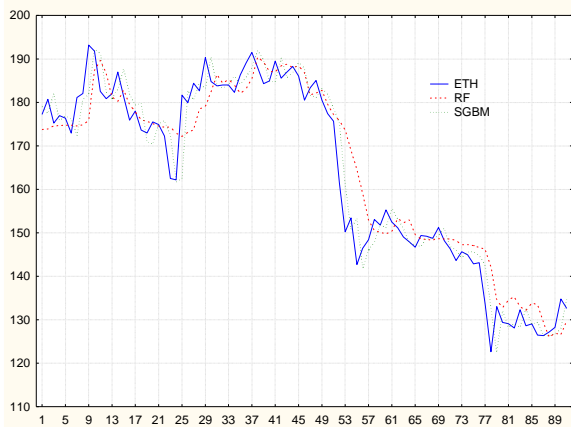


Figure 3. Out of sample prediction of ETH prices

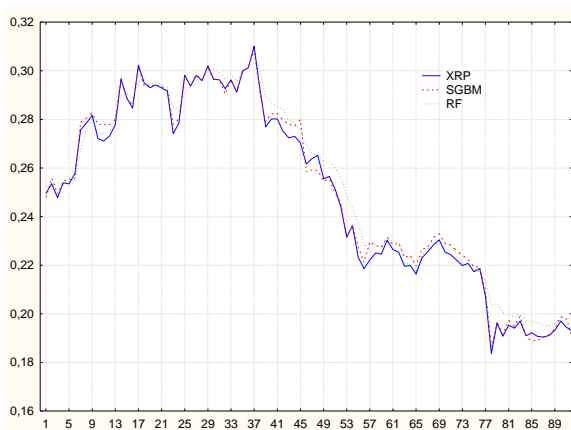


Figure 4. Out of sample prediction XRP prices

accurate. The best prediction performance is produced by SGBM for XRP– 0.92 %, and the best obtained result by RF is 1.84% for XRP.

Our results are comparable in MAPE accuracy metrics with the other research. In [28], the authors obtained prediction error for Bitcoin close prices (MAPE) within 1-2% by using SVM and ANN.

TABLE 2. Out-of-sample accuracy forecasting performance results

	SGBM		RF	
	MAPE, %	RMSE	MAPE, %	RMSE
BTC	2.31	263.34	2.61	305.95
ETH	3.02	5.02	2.26	6.72
XRP	0.92	0.0029	1.84	0.0057

6. CONCLUSION AND DISCUSSION

Our research has shown the efficiency of using ML ensemble-based approaches for predicting cryptocurrency time series. According to our results, the out of sample accuracy of short-term forecasting daily prices obtained by SGBM and RF in terms of MAPE for three of the most capitalized cryptocurrencies (BTC, ETH, and XRP) was within 0.92-2.61 %.

By designing models, we explored different sets of features. Our base dataset contained only past values of the target variable with 14, 21 and 28 lag depth. In this case a larger dataset provided better training of the SGBM and RF, and gave more efficient results. The inclusion of additional features into the model, in particular technical indicators (moving averages), and trading volumes, led to an increased accuracy (MAPE) both on in- and out of samples on average from 1 to 3%.

In our opinion, forecasting accuracy can be improved by including additional features, for example, open, max, min and average prices, fundamental variables, different indicators and oscillators, such as Price rate-of-change, Relative strength index, and so on.

Future research should extend by investigating the predictive power of both features described above and others additional features. In the conclusion, we note that the proposed methodology by the development of combined ensemble of C&RT with other powerful ML models, such as NN and SVM is a promising approach to forecasting not only time series of cryptocurrencies, but also other financial time series. Moreover, it seems promising to us to use DL approaches for feature selection.

7. REFERENCES

1. Coinmarketcap, *Crypto-Currency Market Capitalizations*. <https://coinmarketcap.com/currencies>. Accessed: 15 May 2020.
2. Krugman P., *Bits and Barbarism*. <http://www.nytimes.com/2013/12/23/opinion/krugmanbits-and-barbarism.html>. Accessed: 15 May 2020.
3. CNBC, *Top Economists Stiglitz, Roubini and Rogoff Renew Bitcoin Doom Scenarios*. <https://www.cnbc.com/2018/07/09/nobel-prize-winning-economist-joseph-stiglitz-criticizes-bitcoin.html>. Accessed: 15 May 2020.

4. Selmi, R., Tiwari, A., Hammoudeh, S., "Efficiency or speculation? A dynamic analysis of the Bitcoin market", *Economic Bulletin*, Vol. 38, No. 4, (2018), 2037-2046. <https://econpapers.repec.org/RePEc:ebl:ecbull:eb-18-00395>. Accessed: 15 May 2020.
5. Cheah, E., Fry, J., "Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of bitcoin", *Economic Letters*, Vol., No. 130, (2015), 32-36. doi: 10.1016/j.econlet.2015.02.029
6. Ciaian, P., Rajcaniova, M., and Kancs, A., "The economics of BitCoin price formation", *Applied Economics*, Vol. 48, No. 19, (2016), 1799-1815. doi: 10.1080/00036846.2015.1109038
7. Catania L., Grassi, S., "Modelling Crypto-Currencies Financial Time-Series", *CEIS Research Paper*, Vol. 15, No. 8, (2017), 1-39. <https://ideas.repec.org/p/rtv/ceisrp/417.html>. Accessed: 15 May 2020.
8. Soloviev, V., Belinskij, A., "Complex Systems Theory and Crashes of Cryptocurrency Market", *Communications in Computer and Information Science*, Vol. 1007, (2019) 276-297. https://link.springer.com/chapter/10.1007/978-3-030-13929-2_14.
9. Flach, P.: *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press. Cambridge, UK, (2012).
10. Sheikhi, S., Kheirabadi, M. T., Bazzazi, A. "An Effective Model for SMS spam Detection using Content-based features and Averaged Neural network", *International Journal of Engineering, Transactions B: Applications*, Vol. 33, No. 2, (2020), 221-228. doi: 10.5829/IJE.2020.33.02B.06
11. Kumar, S., Sahoo, G. A., "Random Forest Classifier based on Genetic Algorithm for Cardiovascular Diseases Diagnosis", *International Journal of Engineering, Transactions B: Applications*, Vol. 30, No. 11, (2017), 1723-1729. doi: 10.5829/ije.2017.30.11b.13
12. Patil, S., Phalle, V., "Fault Detection of Anti-friction Bearing using Ensemble Machine Learning Methods", *International Journal of Engineering, Transactions B: Applications*, Vol. 31, No. 11, (2018), 1972-1981. doi: 10.5829/ije.2018.31.11b.22
13. Makridakis, S., Spiliotis, E., Assimakopoulos, V., "Statistical and Machine Learning forecasting methods", *PLoS one*, March 27, (2018), 1-26. doi: 10.1371/journal.pone.0194889
14. Bontempi, G., Taieb, S., Borgne, Y., "Machine Learning Strategies for Time Series Forecasting", European Business Intelligence Summer School eBISS 2012, 62-77. Springer-Verlag. Berlin Heidelberg, (2013). doi: 10.1007/978-3-642-36318-4_3
15. Persio, L., Honchar, O., "Multitask machine learning for financial forecasting", *International Journal of Circuits Systems and Signal Processing*, Vol. 12, (2018), 444-451.
16. Kourentzes, N., Barrow, D.K., Crone, S.F., "Neural network ensemble operators for time series forecasting", *Expert Systems with Applications*, Vol. 41, No. 9, (2014), 4235-4244. doi:10.1016/j.eswa.2013.12.011
17. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE., "A survey of deep neural network architectures and their applications" *Neurocomputing*, Vol. 234 (C), (2017), 11-26. doi:10.1016/j.neucom.2016.12.038.
18. McNally, S., Roche, J., Caton, S. "Predicting the price of bitcoin using machine learning", in: 2018 26th Euromicro Int. Conf. Parallel, Distrib. Network-Based Process., IEEE, (2018), 339-343. doi: 10.1109/PDP2018.2018.00060
19. Hamid SA, Habib A., "Financial Forecasting with Neural Networks", *Academy of Accounting and Financial Studies Journal*, Vol. 18, No. 4, (2014), 37-55. <https://www.abacademies.org/articles/aafsjvol1842014.pdf>. Accessed: 15 May 2020.
20. Boyacioglu, M., Baykan, O.K., "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange", *Expert Systems with Applications*, Vol. 38, No. 5, (2011), 5311-5319. doi: 10.1016/j.eswa.2010.10.027
21. Varghade, P., Patel, R., "Comparison of SVR and Decision Trees for Financial Series Prediction", *International Journal on Advanced Computer Theory and Engineering*, Vol. 1, No. 1, (2012), 101-105.
22. Kumar, M., "Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest", *SSRN Working Paper*, (2006). doi: 10.2139/ssrn.876544
23. Peng, Y., Henrique, P., Albuquerque, M., "The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression", *Expert Systems with Applications*, Vol. 97, (2018), 177-192. doi: 10.1016/j.eswa.2017.12.004
24. Ahmed NK, Atiya AF, Gayar NE, El-Shishiny H., "An Empirical Comparison of Machine Learning Models for Time Series Forecasting", *Econometric Reviews*, Vol. 29, No. 5-6, (2010), 594-621. doi:10.1080/07474938.2010.481556
25. Akyildirim, E., Goncu, A., Sensoy, A., Prediction of Cryptocurrency Returns using Machine Learning, (2018). <https://arxiv.org/pdf/1805.08550.pdf>. Accessed 15 May 2020.
26. Amjad, M., Shah, D., "Trading Bitcoin and Online Time Series Prediction", *NIPS 2016: Time Series Workshop*, (2016). <http://proceedings.mlr.press/v55/amjad16.pdf> (2016). Accessed 15 May 2020.
27. Hitam, N. A., Ismail, A. R., "Comparative Performance of Machine Learning Algorithms for Cryptocurrency Forecasting", (2018). <https://www.researchgate.net/publication/327415267>. Accessed 15 May 2020.
28. Mallqui, D., Fernandes, R. "Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques", *Applied Soft Computing Journal*, Vol. 75, (2019), 596-606. doi: 10.1016/j.asoc.2018.11.03
29. Sezer, O.B., Mehmet Ugur Gudelek, M.U., Ozbayoglu, A.M. "Financial Time Series Forecasting with Deep Learning : A systematic literature review: 2005-2019", *Applied Soft Computing Journal*, Vol. 90, (2020), 106-181. doi: 10.1016/j.asoc.2020.106181
30. Kumar, D., Rath, S.K. "Predicting the Trends of Price for Ethereum Using Deep Learning Techniques", in: Artificial Intelligence and Evolutionary Computations in Engineering Systems, Springer, 2020, 103-114. doi: 10.1007/978-981-15-0199-9_9
31. Yao, Y., Yi, J., and Zhai, S., "Predictive Analysis of Cryptocurrency Price Using Deep Learnin", *International Journal of Engineering & Technology*, Vol. 7, No. 3, 27, (2018), 258-264. doi: 10.14419/ijet.v7i3.27.17889
32. Chen, W., Xu, H., Jia, L., Gao, Y. "Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants", *International Journal of Forecasting*, (2020), (article in press). doi: 10.1016/j.ijforecast.2020.02.008
33. Saxena, A., Sukumar, T., "Predicting bitcoin price using LSTM and compare its predictability with ARIMA model", *International Journal of Pure and Applied Mathematics*, Vol. 119, No. 17, (2018), 2591-2600. <https://acadpubl.eu/hub/2018-119-17/3/214.pdf>. Accessed: 15 May 2020.
34. Breiman, L., Friedman, H., Olshen, R. A., & Stone, C. J., *Classification and Regression Trees*. Belmont, NJ. Wadsworth International Group, (1984).

35. Breiman, L., "Random Forests", *Machine Learning*, Vol. 45, (2001), 5-32. doi: 10.1023/A:1010933404324
36. Friedman, Jerome H., "Greedy Function Approximation. A Gradient Boosting Machine", *Annals of Statistics*, Vol. 29, No. 5, (2001), 1189-232. doi: 10.2307/2699986
37. Friedman, Jerome H., "Stochastic Gradient Boosting", *Computational Statistics and Data Analysis*, Vol. 38, No. 4, (1999), 367-378. doi: 10.1016/S0167-9473(01)00065-2
38. Borges, T.A., Neves, R.N. "Ensemble of Machine Learning Algorithms for Cryptocurrency Investment with Different Data Resampling Methods", *Applied Soft Computing Journal*, Vol. 90, (2020), 106-187. doi: 10.1016/j.asoc.2020.106187
39. Chen, Z., Li, C., Sun, W. Bitcoin Price Prediction Using Machine Learning: An Approach to Sample Dimension Engineering, *Journal of Computational and Applied Mathematics*, Vol. 365, (2020), 112395. doi: 10.1016/j.cam.2019.112395
40. Sun, X., Liu, M., Sima, Z. "A novel cryptocurrency price trend forecasting model based on LightGBM", *Finance Research Letters*, Vol. 32, (2020), 101084. doi: doi.org/10.1016/j.frl.2018.12.032
41. Yahoo Finance. <https://finance.yahoo.com>. Accessed: 15 May 2020.
42. Guryanova, L., Yatsenko, R., Dubrovina, N., Babenko, V. "Machine Learning Methods and Models, Predictive Analytics and Applications". Machine Learning Methods and Models, Predictive Analytics and Applications 2020: Proceedings of the Workshop on the XII International Scientific Practical Conference Modern problems of social and economic systems modelling (MPSESM-W 2020), Kharkiv, Ukraine, June 25, 2020, Vol-2649, (2020), 1-5. URL: <http://ceur-ws.org/Vol-2649/>
43. Alessandretti, L., ElBahrawy, A., Aiello, L., Baronchelli, A., "Anticipating Cryptocurrency Prices Using Machine Learning", *Hindawi Complexity*, (2018) doi: 10.1155/2018/8983590

Persian Abstract

چکیده

در این مقاله مشکلات پیش بینی کوتاه مدت سری زمانی ارز رمزنگاری شده با استفاده از یک رویکرد یادگیری ماشین تحت نظارت ML بحث شده است. برای این منظور، ما از روش قدرتمندترین گروه از جمله جنگل های تصادفی (RF) و دستگاه تقویت گرادیان تصادفی SGBM استفاده کردیم. همانطور که مجموعه داده از قیمت بسته شده روزانه سه سکه با بیشترین سرمایه جمع آوری می شود: بیت کوین BTC، اتریوم ETH و ریپل XRP، و به عنوان ویژگی هایی از اطلاعات قیمت گذشته و شاخص های فنی (میانگین متحرک) استفاده کردیم. برای بررسی اثربخشی این مدل ها، ما با استفاده از تکنیک یک قدم جلوتر پیش بینی خارج از نمونه برای سری های زمانی انتخاب شده را انجام دادیم. میزان دقت قیمت پیش بینی شده با استفاده از RF و GBM محاسبه شد. نتایج به کارگیری رویکرد گروه های ML برای پیش بینی قیمت ارزهای رمزنگاری شده را تأیید می کند. صحت خارج از نمونه پیش بینی کوتاه مدت قیمت های روزانه بسته شده توسط SGBM و RF از نظر میانگین درصد مطلق خطا (MAPE) برای سه ارز رمز پایه (BTC، ETH و XRP) در محدوده ۰٫۹۲-۲٫۶۱٪ بود.
