Forecasting of Cryptocurrency Prices Using Machine Learning

Vasily Derbentsev, Andriy Matviychuk, and Vladimir N. Soloviev

Abstract Our study is devoted to the problems of the short-term forecasting cryptocurrency time series using machine learning (ML) approach. Focus on studying of the financial time series allows to analyze the methodological principles, including the advantages and disadvantages of using ML algorithms. The 90-day time horizon of the dynamics of the three most capitalized cryptocurrencies (Bitcoin, Ethereum, Ripple) was estimated using the Binary Autoregressive Tree model (BART), Neural Networks (multilayer perceptron, MLP) and an ensemble of Classification and Regression Trees models—Random Forest (RF). The advantange of the developed models is that their application does not impose rigid restrictions on the statistical properties of the studied cryptocurrencies time series, with only the past values of the target variable being used as predictors. Comparative analysis of the predictive ability of the constructed models showed that all the models adequately describe the dynamics of the cryptocurrencies with the mean absolute persentage error (MAPE) for the BART and MLP models averaging 3.5%, and for RF models within 5%. Since for trading perspective it is of interest to predict the direction of a change in price or trend, rather than its numerical value, the practical application of BART model was also demonstrated in the forecasting of the direction of change in price for a 90-day period. To this end, a model of binary classification was used in the methodology for assessing the degree of attractiveness of cryptocurrencies as an innovative financial instrument. Conducted computer simulations have confirmed the feasibility of using the machine learning methods and models for the short-term forecasting of financial time series. Constructed models and their ensembles can be the basis for the algorithms for automated trading systems for Internet trading.

V. Derbentsev · A. Matviychuk
Kyiv National Economic University named after Vadym Hetman, 54/1 Prospect Peremogy, Kyiv 03057, Ukraine
e-mail: derbv@kneu.edu.ua

A. Matviychuk
e-mail: editor@nfmte.com

V. N. Soloviev (✉)
Kryvyi Rih State Pedagogical University, 54 Gagarina Ave, Kryvyi Rih 50086, Ukraine
e-mail: vnsoloviev2016@gmail.com

L. Pichl et al. (eds.), *Advanced Studies of Financial Technologies and Cryptocurrency Markets*, https://doi.org/10.1007/978-981-15-4498-9_12

Keywords Binary autoregressive tree model · Cryptocurrency prices · Financial time series · Machine learning · Neural network · Regression and classification tree ensemble · Short-term forecasting

1 Introduction

Current stage of the global development has been characterized by the widespread Information Technology (IT) innovation in all spheres of human activity, especially in business and finance. Probably, today the question about the role and prospects of widespread implementation of the blockchain technology and the first cryptographic currency (cryptocurrency) Bitcoin, which was developed in 2009, is the most controversial.

This problem is the focus of debate among leading economists, politicians and businessmen, whose views are often diametrically opposite: from full support ("digital gold" of the twenty-first Century and the future of the world currency reserve (Popper 2015; Vigna and Casey 2015)), to complete negation ("financial bubble", the biggest financial shady transaction (Krugman 2013; CNBC 2018)).

This controversy is not least due to the significant fluctuations in the exchange rate of cryptocurrencies and legal uncertainty of the transactions with them in most countries of the world, which led to significant risks of investment in these assets.

In this regard, the problem of developing adequate cryptocurrency prices forecasting approach is relevant to the scientific community as well as to financial analysts, investors and traders.

In order to make investment decisions in the crypto market, it is necessary to have efficient tools of prices forecasting, profitability and risk assessment, at least for the shortterm time horizon.

Analysis of recent theoretical and empirical studies shows that the price dynamics of cryptocurrencies are influenced by many latent factors. These key factors (drivers) have not been well understood and identified yet (Selmi et al. 2018; Cheah 2015; Ciaian 2016; Catania and Grassi 2017). The vast majority of researchers are inclined to believe that the fundamental factors do not have a significant influence on the cryptocurrency rate. Instead their prices are determined by the demand-supply ratio.

In our recent studies, we used the methods of the complex systems theory and demonstrated the possibility of constructing indicators of critical and crash phenomena in the volatile stock and cryptocurrency markets (Derbentsev et al. 2019b; Soloviev and Belinskij 2016, 2019; Soloviev et al. 2019a, b, c; Belinskyi et al. 2019). Our results show that cryptocurrency time series are characterized by complex dynamics, extreme observations and a high degree of volatility. They are also non-stationary, fractal and have non-Gaussian distributions (Belinskyi et al. 2019). These results are consistent with several other empirical studies which applied the statistical approach (Catania and Grassi 2017).

Therefore, the application of traditional forecasting methods based on the use of casual models, built within a certain theoretical macroeconomic concept, or classical time series models has proven to be ineffective.

In the last two decades the methods and algorithms of machine learning have been applied to forecasting financial and economic time series (Flach 2012; Bontempi et al. 2013; Persio and Honchar 2018), and various automated trading systems—bots built on these algorithms—began to be used for trading.

The main purpose of our research is to compare the prognostic properties for the short-term prediction task of the cryptocurrency exchange rates of several ML methods: the BART algorithm (Derbentsev et al. 2019a), Artificial Neural Networks (ANN) and decision trees ensemble—RF.

The paper is structured as follows. Section 2 describes previous studies in these fields. Section 3 presents ML approach in the context of financial time series forecasting. In this section we described the main aspects of applying BART, ANN and RF to prediction of cryptocurrency prices.

Section 4 describes the datasets used to test and simulation the models. The empirical results are reported in Sect. 5. In this section we presented the results of the short-term predictions obtained with BART, ANN and RF models for the prices of the three most capitalized cryptocurrencies (Bitcoin (BTC), Ethereum (ETH) and Ripple (XRP)), and their price direction changes. And finally, we discuss results of our study in Sect. 6.

2 Analysis of Previous Studies

Recently non-parametric methods within the Machine Learning (ML) and Deep Learning (DL) paradigms have been widely used for predicting financial time series, in particular, cryptocurrency prices dynamics (Varghade and Patel 2012; Boyacioglu and Baykan 2011; Okasha 2014; Kumar 2006; Peng et al. 2018; McNally 2016).

In this area the primary focus has been on the use of such methods as ANNs of different types and architectures, and Support Vector Machines (SVM). The application of these methods has proven to be more efficient for the forecasting tasks for both “traditional” (fiat currency, stock indices, commodities prices, etc.) (Varghade and Patel 2012; Boyacioglu and Baykan 2011; Okasha 2014; Kumar 2006) and innovative financial assets, including cryptocurrencies (Peng et al. 2018; McNally 2016; Saxena and Sukumar 2018; Amjad and Shah 2016; Alessandretti et al. 2018).

Thus, examples of effective use of SVM in forecasting volatility of fiat- and cryptocurrencies are given, in particular, by Peng et al. (2018).

Several studies (McNally 2016; Saxena and Sukumar 2018; Amjad and Shah 2016) presented the results of BTC exchange rate prediction by using ARIMA models, RF, Logistic Regressions (LR), Linear Discriminant Analysis approach (LDA) and such ANN as Long Short-Term Memory (LSTM). According to obtained results, the ML models proved to be more accurate in prediction both cryptocurrency prices, and their volatility than times series models.

Rebane and Karlsson (2018) presented a comparative analysis of the prognostic properties of ARIMA with Recurrent Neural Networks (RNN) for such cryptocurrencies as Bitcoin, DASH, Ethereum, Litecoin (LTC), Siacoin (SC), Stellar (STR), NEM (XEM), Monero (XMR) and Ripple (XRP). Their results also revealed better predictive properties of ANN than ARIMA models.

Comparative performance of ML algorithms for forecasting cryptocurrency prices has reported in the paper of Hitam and Ismail (2018). They tested ANNs, SVM and Deep Learning (Boosted NN) for such coins as BTC, ETH, LTC, XEM, XRP and XLM. Their results show that SVM has the best predictive accuracy in the terms of the lowest value of Mean Percentage Error.

Yao et al. (2018) proposed to predict cryptocurrency price by using more a wider dataset, which includes not only prices, but also market cap, volume, circulating and maximum supply. Based on their results obtained on deep learning techniques (RNN and LSTM) the prediction accuracy was within 59% (when using only prices) and up to 75% (on an extended dataset).

Another powerful class of ML methods are the Classification and Regression Tree (C&RT) and their ensembles proposed by Leo Breiman and colleagues (Breiman et al. 1984; Breiman 2001). It should be noted that much less attention has been paid to these algorithms in the field of modelling and forecasting financial times series (see, for example (Varghade and Patel 2012; Kumar 2006)).

In our recent work (Derbentsev et al. 2019a), we proposed BART algorithm, which is a generalization of C&RT models for the case of scalar time series. The application of BART to cryptocurrency exchange rate prediction task demonstrated that is was more efficient than the ARIMA-ARFIMA time series models.

Nowadays combined classical econometric methods as well as methods of machine learning (Albuquerque et al. 2018; Wang et al. 2018) and those which take into consideration the spirit of social networks regarding the state and tendency of cryptocurrency dynamics (Kennis 2018) are becoming more popular.

Another important aspect in the forecasting dynamics of financial time series is prediction of the price changes direction. For this purpose Kumar (2006) tested such ML classification models as LDA, LR, ANN, RF and SVM. His empirical results suggests that the SVM and RF outperforms the other classification methods for the prediction direction of the stock market movement.

Akyildirim et al. (2018) investigated predictability of the 12 cryptocurrencies on the both daily and minute datasets by using the ML classification algorithms (SVM, LR, ANN and RF) with the past price information and technical indicators as model features. Their results showed that the direction of returns in the cryptocurrency market can be predicted with averages accuracy around 55–60% with the daily or minute observation.

In our previous works (Matviychuk 2006, 2011) we also solved the problem of prediction of the price changes direction of financial time series. To this end we applied the Fuzzy Logic tools were for formation of a knowledge base we used rules of wave development from technical analysis and Elliott wave theory. And also the task of pattern recognition in the structure of price curves and prediction of their further development we had dealt with usage of Counterpropagation Neural Networks.

3 Methodology

3.1 Machine Learning Approach of Forecasting Cryptocurrency Prices

The main difference between ML and classical modeling is that the Machine Learning algorithms interpret the data themselves, so there is no need to perform their initial decomposition. Depending on the purpose of the analysis, these algorithms "build" logic modeling based on the available data. This avoids the complex and lengthy pre-model stage of statistical testing of various hypotheses.

The main purpose of our study is to determine the ability of ML methods to effectively analyse the time series data of cryptocurrencies (both scalar and vector), and to identify the patterns and time correlations that form the basis for the qualitative forecasts.

An important characteristic of ML is that the methods used to search for templates in the data do not imply a priori data structure, their statistical properties and the type of relationships.

Within the ML paradigm, a number of powerful approaches, methods and algorithms have been developed, such as ANNs, SVM, C&RT, RF Regression and Classification ensembles, Gradient Boosting (GBoost), Deep Neural Networks and Deep Learning, Kernel methods, etc. (Flach 2012).

Among ML methods, neural networks of different architecture, particularly deep networks, have gained the most popularity. Numerous empirical studies have shown the effectiveness of the application of ANN to pattern recognition, image and voice analysis, machine translation, etc. They are increasingly being used to analyse and forecast financial time series, in particular cryptocurrency data.

Several studies (Boyacioglu and Baykan 2011; Hitam and Ismail 2018; Matviychuk 2011) showed that ANNs have better predictive properties than time series models and other ML algorithms for financial time series forecasting.

Another type of ML models is C&RT and their ensembles. Both ANN and C&RT approaches have their own advantages and disadvantages. Their common advantages are the following:

- they do not impose strict a priori assumptions about the input data;
- they have a high level of automation, because required mathematical tools are built into majority data mining software;
- they are able to process data both quantitative (metric) and qualitative (categorical).

The common disadvantages of both ANN and C&RT are the overfitting problem, and a large number of hyperparameters that require tuning. The overfitting leads to significantly increasing forecast errors on new data.

As for ANNs, they are "Black Box" model which are characterized by the "opacity" of the hypothesis function (a function that describes the relationship between input and output). So ANNs don't have enough explanatory power and they require significant training time. In addition, choosing a network architecture, the number of input neurons, hidden layers and activation functions is generally a non-trivial task.

The major weaknesses of the C&RT models are their lower accuracy compared to ANNs (for the regression problems) and the ambiguity of choosing the best final tree (for the prediction problem). But their advantage is visibility, perspicuity for visualization and interpretation.

However, complex tree branches are also difficult to interpret in a meaningful way, therefore, using them, we have to find a compromise between the complexity of the tree and its accuracy. This problem is inherent in the vast majority of ML algorithms.

The RF algorithm consists of constructing an ensemble of simple classifiers (trees) and obtaining an average estimate of the prediction of each of the trees that are built on different subsets of features and randomly selected training subsamples of data. This approach is less subject to overfitting, but is also poorly interpreted.

The input data for our analysis is a time series of values for a certain cryptocurrency of length T, which we denote by $Y = (Y_1, Y_2, \ldots, Y_T)$. We will use supervised learning, so training and test samples contain a set of examples. In our case this is one-step ahead forecast Y_t with known values of the target variable in p previous time periods $Y_{t-1}, Y_{t-1}, \ldots, Y_{t-p}$.

We state our hypothesis in the following form

$$\Pr(Y_t | Y_1, Y_2, \ldots, Y_{t-1}, \theta) = f\left(Y_t | Y_{t-p}, Y_{t-p+1}, \ldots, Y_{t-1}, \theta\right), \quad p < t \leq T, \tag{1}$$

where $f(\cdot|\cdot, \theta)$—is a family of conditional probability distributions, and θ—are unknown model parameters.

The hypothesis function can be represented as

$$\widehat{Y}_t = \hat{f}\left(Y_{t-p}, Y_{t-p+1}, \ldots, Y_{t-1}, \theta\right) + \varepsilon_t. \tag{2}$$

Thereby we used only past values of the target variable as factors (features).

We investigated three different type of ML algorithms to predict cryptocurrency time series (short-term forecast) and compare their predictive properties: the Binary Auto Regressive Tree, the Multilayer Perceptron, and the Random Forest tree ensemble models.

It should be mentioned that when we applying ML methods, 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:

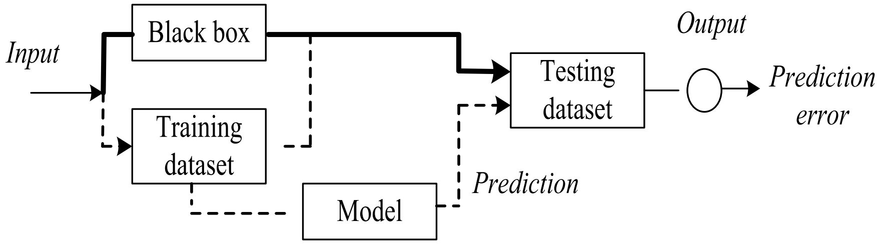

Fig. 1 Supervised machine learning prediction

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

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):

$$PE(Y_t) = \mathrm{E}\left[\left(Y_t - \hat{f}(Y_t)\right)^2\right] = \mathrm{Bias}^2\left(\hat{f}\right) + \mathrm{Var}\left(\hat{f}\right) + \sigma^2, \tag{3}$$

where $PE(Y_t)$—the total forecast error at time t; $\mathrm{E}(\cdot)$—mathematical expectation operator; $Y_t, \hat{f}(Y_t)$—the actual time series value and its predicted value; $\mathrm{Bias}(\cdot)$—the average bias across all datasets; $\mathrm{Var}(\cdot)$—error variance, which generally depends on the number of model parameters and their accuracy; σ^2—unavoidable error.

A general diagram of supervised ML prediction process is shown in Fig. 1.

3.2 Binary Auto Regressive Tree (BART)

Binary Auto Regressive Tree is a generalization of standard C&RT models, which is adapted to time series prediction tasks. BART combines the classic C&RT algorithm (Kumar 2006) and the ARIMA Box-Jenkins autoregressive models.

The target variable Y_t in this algorithm depends on p the previous values of the studied time series $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$. BART allows dividing the phase space into segments, with a subsequent development of a model for each, and a piecewise regression function presented in an intuitive and visual way. In such a tree, the inner nodes contain rules for splitting the space of explanatory variables; branches indicate conditions and transition between nodes; and the leaves are local ARIMA models.

When constructing BART a binary tree is constructed, therefore each node has two child nodes (i.e., number of branches is 2). An autoregressive tree is constructed sequentially (iteratively) and this process is described by the following algorithm (Derbentsev et al. 2019a; Breiman et al. 1984).

Step 1. The first step is to determine the threshold for splitting the initial (root) node, which is taken as the median Me (2-quantile $Q_{50\%}$) of the training series (sample) and is calculated by the formula

$$Me(Y) = Q_{50\%} = 0.5 \times \left(Y^{\min} + Y^{\max}\right), \tag{4}$$

The median of the time series is defined as the median of the distribution of the realization of a random variable at time t. For a stationary time series (or time series with a symmetric distribution), this value is independent of the observation time and then the sample median is equal to mean, i.e. $Me(Y) = \overline{Y}$.

Therefore, an autoregressive estimation of the tree at the first step of splitting will look like

$$f(Y_t) = Me(Y) I_R(Y_{t-1}), \tag{5}$$

where R is the dataset; $I_R(Y_{t-1})$—an indicator function of space, in fact it is a set of rules for getting variable Y_{t-1} into this space. So, in the first step, the dataset is divide into two subsets by criterion (5).

Step 2. The second step is to divide the data space in the selected node obtained in the first step into two parts. Some lag variable, for example, $Y_{t-k}, k \in (1, 2, \ldots, p)$ is selected and the left and right data subspaces R_{left}, R_{right} are defined:

$$R_{left} = \{Y_{t-k} \in R : Y_{t-k} \leq \alpha\}, R_{right} = \{Y_{t-k} \in R : Y_{t-k} > \alpha\}, p < t \leq T. \tag{6}$$

Then the regression estimation at the next step takes the form:

$$f(Y_t) = \left(\frac{1}{M} \sum_{I_1} Y_{t-k}^{(i)}\right) I_{R_{left}}(Y_{t-k}) + \left(\frac{1}{N} \sum_{I_2} Y_{t-k}^{(i)}\right) I_{R_{right}}(Y_{t-k}), \tag{7}$$

where $I_1 = \left\{i,\ Y_{t-k}^{(i)} \in R_{left}\right\}, I_2 = \left\{i,\ Y_{t-k}^{(i)} \in R_{right}\right\}$—sets of observation indices (i) falling into the subspaces R_{left} and R_{right} respectively; M, N are the number of elements in these subspaces.

Estimation of the best split is equal to the smallest sum of squares

$$R\left(\hat{f}\right) = \frac{1}{T} \sum_{t=p+1}^{T} \left(Y_t - \hat{f}\left(Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}\right)\right)^2. \tag{8}$$

Step 3. For each untreated node, the best splitting is found. There are two arguments defined for this: the variable $Y_{t-k}, k \in (1, 2, \ldots, p)$ that will be splitting and the threshold value α of this variable.

We used as a threshold quintile the corresponding empirical distribution of the random variable Y (the value which random variable does not exceed with a certain probability) and limited the potential splitting on seven values of each predictor variable

$$\alpha \in \{Q_{10\%}, Q_{25\%}, Q_{40\%}, Q_{50\%}, Q_{60\%}, Q_{75\%}, Q_{90\%}\}. \tag{9}$$

Of the possible splitting options in this step, the "better" option is chosen by the adopted rule. These procedures are similar to the C&RT algorithm (Breiman et al. 1984). The difference is in the adopted rules, evaluation criteria and stop splitting. BART suggested an alternative criterion for selecting the best splitting based on the entropy (called Entropy Information Gain, IGain), because this reduces the complexity of the tree

$$IGain = \widehat{H}\,(M, N) - \widehat{H}\,(m, n), \tag{10}$$

where $\widehat{H}\,(M, N)$ is entropy of parent node, $\widehat{H}\,(m, n)$ is average entropy of children nodes.Thus, for each next splitting, algorithm selects node and lag variable (and, accordingly, the threshold value) that provide the maximum entropy reduction given by (10).

Step 4. In the next step it is necessary to evaluate the "value" of the tree, which characterizes the relationship between the accuracy of the approximation and the complexity (branching) of the constructed tree.

The value of the tree in BART is determined based on the early stop criterion. As such criterion, we used the Extended Bayesian Information Criterion (EBIC), which minimizes statistics:

$$EBIC = T \cdot \ln R\left(\hat{f}\right) + J \cdot [\ln(T) + 2\ln(b)], \tag{11}$$

where $R(\hat{f})$ is the root mean square error (8); J—number of model settings; T—number of samples in training set; b is the quantity that characterizes the complexity of the model space. It equals the product of the size of the tree (the number of branches in the tree) by the number of lag variables p.

In expression (11), the first term is the maximum value of the logarithmic function of the root mean square error, and the second is a penalty for the complexity of the model.

Step 5. Splitting nodes continues as long as the value of statistics EBIC decreases. If the selected splitting is effective at entropy gain (11), then it must be performed and the algorithm proceeds to step 3 (to evaluate other nodes). Otherwise, the final tree is selected and the BART algorithm is completed.

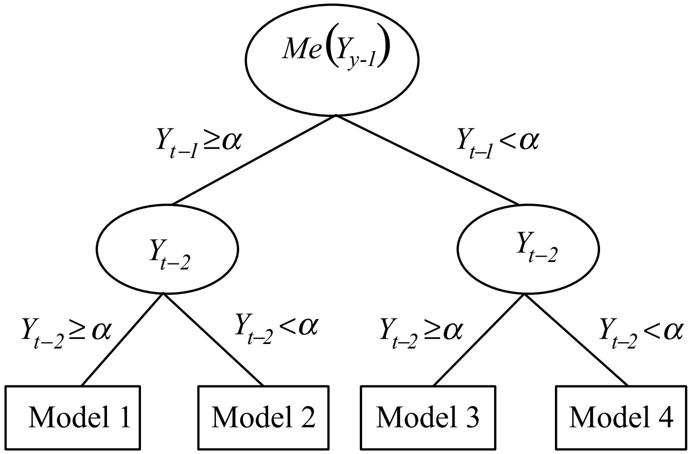

Fig. 2 Example of building BART with 2 split variables ($p = 2$)

Because the final target of the algorithm is prediction, we proposed to build Box-Jenkins ARIMA models on the each leaf nodes.

Fig. 2 shows a simple example of building BART with 2 split variables ($p = 2$), with local AIMA models located on leaf nodes.

Each of these models approximate thier own phase sub-space factor variables.

3.3 Random Forest

The random forest algorithm is based on the construction of an ensemble of classification (regression) trees, each of which is constructed from sub-samples of the original training sample using *bagging* (abbreviated from **b**ootstrap **agg**regat**ing**) (Breiman 2001). Bagging is a method of creating an ensemble of models based on various random samples from the original dataset. Samples are uniformly replaced and are called *bootstrap samples* (Flach 2012).

Bagging efficiency is achieved by training the basic algorithms in different sub-sets. These sub-sets will be significantly different from each other, and their errors are mutually compensated by "voting", as well as anomalous observations and time series jumps may not be included in some training sub-sets.

Bagging is especially useful in combination with tree models that are sensitive to changes in training data. In the RF algorithm, bagging is combined with the method of random subspaces: that is, each tree is built on different randomly selected subsets of features—this process is called *subspace sampling*.

The random subspaces method reduces the correlation between trees and avoids retraining because the basic algorithms are trained on different subsets of traits, which are also randomly selected.

As a result, the diversity of the ensemble will be even greater, reducing the learning time of each tree, which can be done in parallel. This ensemble is called a Random Forest.

The RF is used for both classification and regression problems, and RF can also be useful for selecting predictors and finding deviations in data analysis.

The prediction with RF algorithm is carried out by averaging the forecasts obtained by each ensemble tree (or by "voting" the trees for classification problems). Unlike individual trees, this algorithm is much less prone to overfitting and gives more sensitive (flexible) boundary to decision making.

3.4 Neural Network

As an ANN model, we used the simplest and most common Multilayer Perceptron architecture with one hidden layer of neurons, and an output layer containing only one neuron—estimation of the forecast of the studied time series by one step (Fig. 3).

According to Kolmogorov's theorem despite such a simple architecture, MLP can describe complex patterns in the data and modeled unknown nonlinear function of the time series with sufficient accuracy. This is achieved by using superposition of nonlinear activation functions on the hidden and output layers of the network.

Network output values depend on input and hidden neurons, weights, and activation functions

$$\widehat{Y}_{t+1} = g\left(\sum_{i=1}^{k} w_i f(s_i) + b_0\right), \tag{14}$$

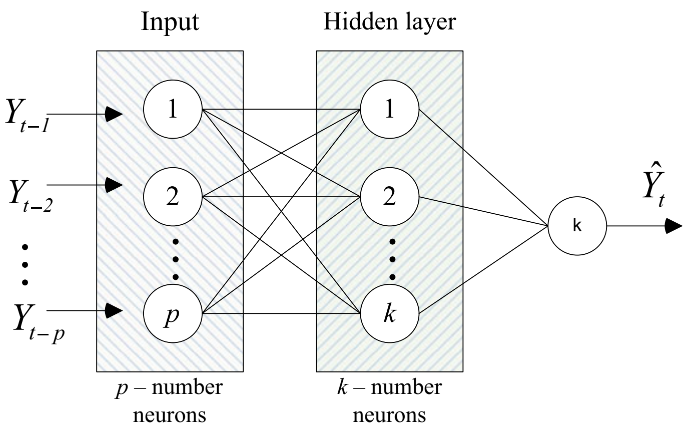

Fig. 3 Multilayer perceptron

where $f(\cdot), g(\cdot)$ are activation functions of the hidden and output layer neurons respectively; w_i—weights of links between hidden layer neurons and the output of the network; b_0, b_i—neurons bias of the output and hidden layers; $s_i = \sum_{j=1}^{p} \omega_{ji} Y_{t-j+1} + b_i$—sum of hidden layer neurons; ω_{ij}—weight of links between neurons of input and hidden layers.

MLP training consists in computing synaptic weights, and Error (Cost) Function (EF) is used to determine the difference between the target variable and the network output. Finding the minimum EF was performed using the gradient descent method.

We used a back-propagation algorithm. According to it the value of the EF is applied to the neurons of the hidden layer and the weights are adjusted. In the first step the input vector $Y_n, Y_{n+1}, \ldots, Y_{n+p}, (n = 1, 2, \ldots, t)$ propagates across the network from layer to layer in the forward direction with the fixed scales. In the next, reverse step, all synaptic weights are adjusted by the error correction rule.

4 Data

For numerical simulation of the short-term forecasting models (BART, RF and MLP) of cryptocurrency prices we selected data of daily exchanges of the three most capitalized coins: Bitcoin, Ethereum and Ripple. Data set includes 1583 observations for the period from August 1, 2015 to December 1, 2019 according to the Yahoo Finance (2019).

We chose closing prices both in absolute value and in natural log, which allows to stabilize the variability (variance) of the studied series (Fig. 4).

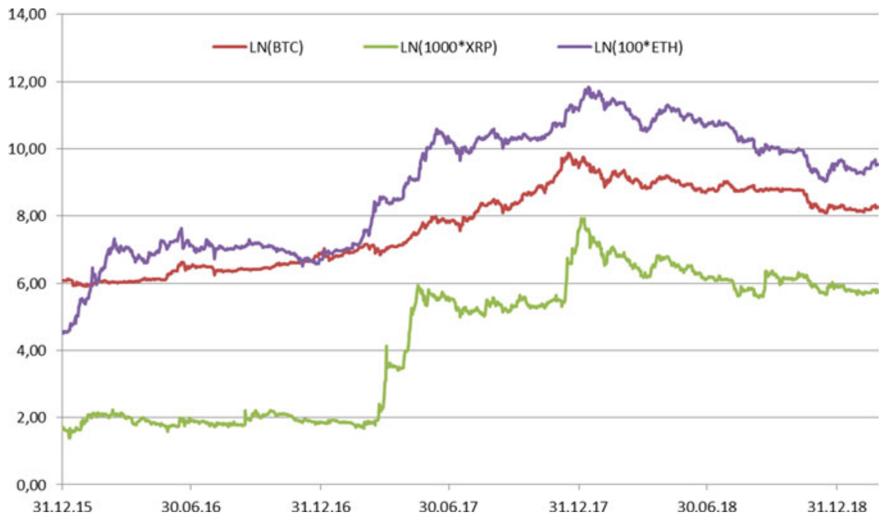

Fig. 4 Daily close prices of BTC, ETH and XRP (USD, log scale)

The first 1392 observations were divided into 80 and 20% between the training and test sets and were used to fit and train models and tuning their parameters, and the last 90 observations were reserved to estimate the quality of the forecast.

5 Empirical Result

Because all three types of models useses only past observations of the time series, the choice of the lag depth *p* is one of the main tasks for identifying them. According to many empirical studies (Boyacioglu and Baykan 2011; Okasha 2014; Matviychuk 2011), for "traditional" financial assets (fiat currencies, stock indices, commodity prices, etc.) that are traded for 5 days a week, there is a seasonal lag which is a multiple of 5 if we use daily observations.

Cryptocurrencies are traded 24/7, that's why it is expected a seasonal lag multiple of 7 days exists. Correlation analysis confirmed our hypothesis: for all 3 cryptocurrencies there are statistically significant correlations on lags 7, 14, 21, besides there are correlations on some other lags. Similar results were obtained in Catania and Grassi (2017), Alessandretti et al. (2018).

We tested 3 classes of models (BART, RF, MLP) with different lag depth for each cryptocurrency.

According to our hypothesis regarding lag depth for MLP models, we tested the following architectures:

- 7 inputs and 4–12 hidden layer neurons;
- 14 inputs and 5–15 hidden layer neurons;
- 21 inputs and 6–21 hidden layer neurons.

The most common functions such as logistic, hyperbolic tan, exponential and ReLu were selected as activation functions. Training MLP for each cryptocurrency and different lag values (number of input neurons) was conducted over 100 epochs, of which the best 5 architectures were selected for each case (in terms of minimum PE error (3) in the test sample and matching the model residuals to normal distribution).

The final prediction for each cryptocurrency was obtained as the prediction of the ensemble of networks, that is, average of the best 5 corresponding MLP models.

For RF simulation we used the following parameter settings: total number of trees—200, maximum tree depth—10, and number of predictors in each tree: 3, 5, 7 for RF-7, RF-14 and RF-21 models respectively.

For BART we chose two parameters: a maximum tree depth—15, a minimum number of examples (observations) per node—20.

Figure 5 shows the graphs that characterize the quality of approximation of BART models for training (a, c, e) and test (b, d, f) samples for 3 cryptocurrencies. The graphs (a, c, e) show the dependence of the predicted values (vertical axis) on the actual data (horizontal axis).

The short-term forecasts for each of the cryptocurrency were made for both absolute values of prices and their logs. It should be noted that according to our results

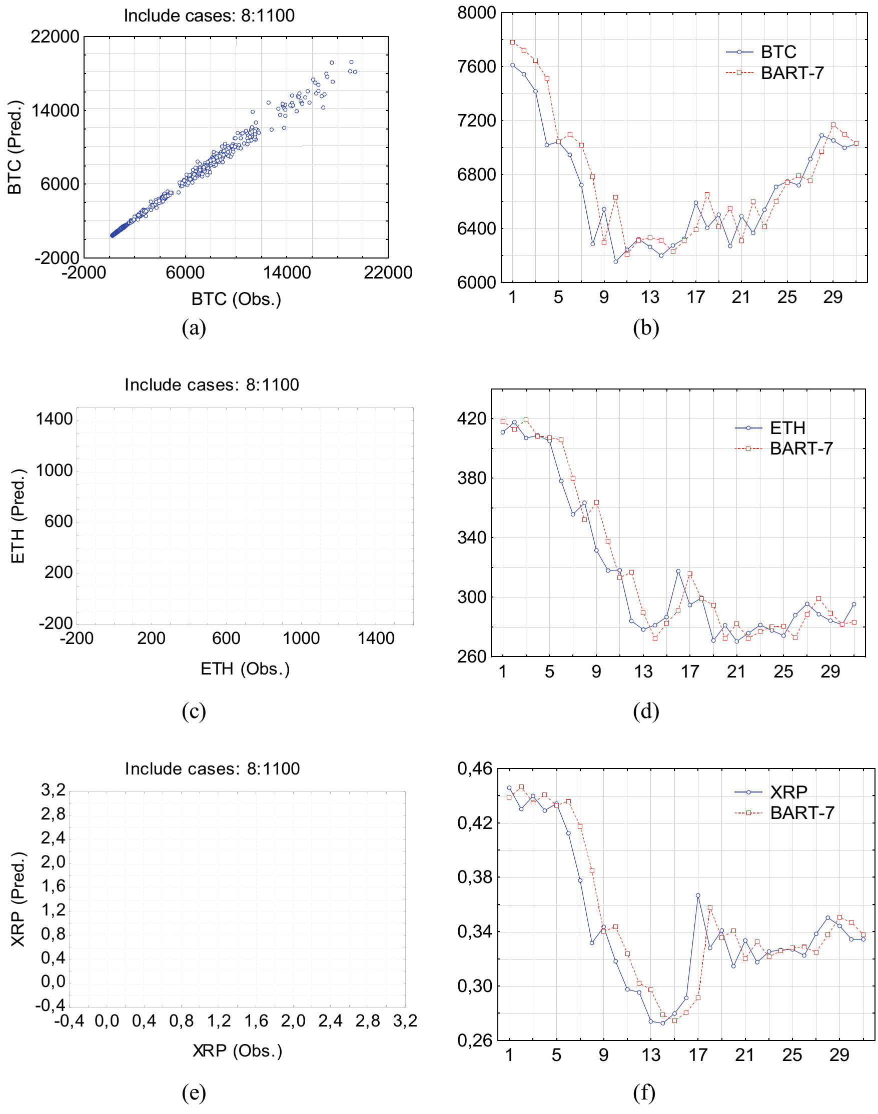

Fig. 5 Quality of approximation of BART models for training and test sets for BTC (**a**, **b**), ETH (**c**, **d**) and XRP (**e**, **f**)

the prediction accuracy by the metrics (15) defined below for the logs of prices was generally no better than for the absolute values.

This fact supports the argument that the ML algorithms (in particular, ANNs, C&RT and their ensembles) are much less sensitive to the time series statistical properties than classical statistical and econometric methods.

Figures 6, 7 and 8 show the final results of forecasting cryptocurrency prices for

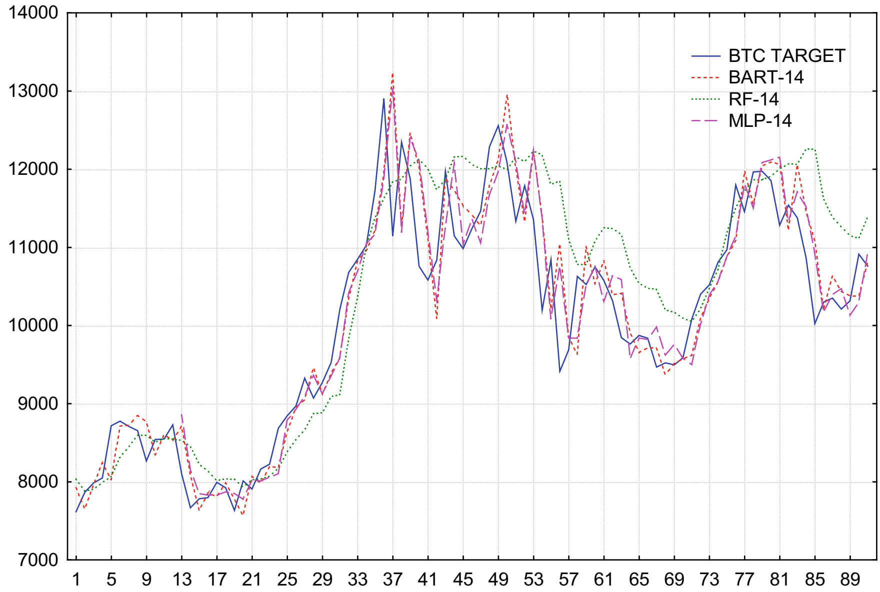

Fig. 6 90-day forecast of BTC prices (lag $p = 14$)

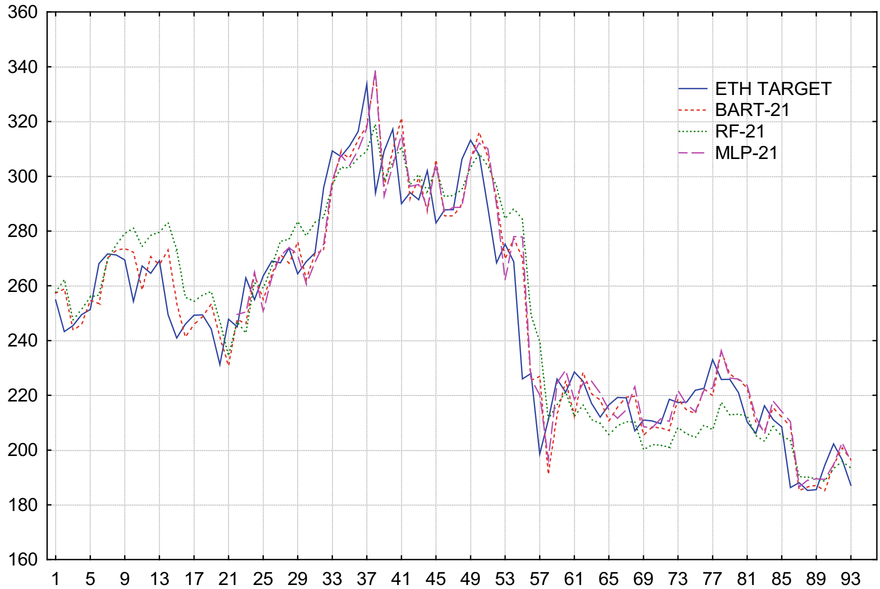

Fig. 7 90-day forecast of ETH prices (lag $p = 21$)

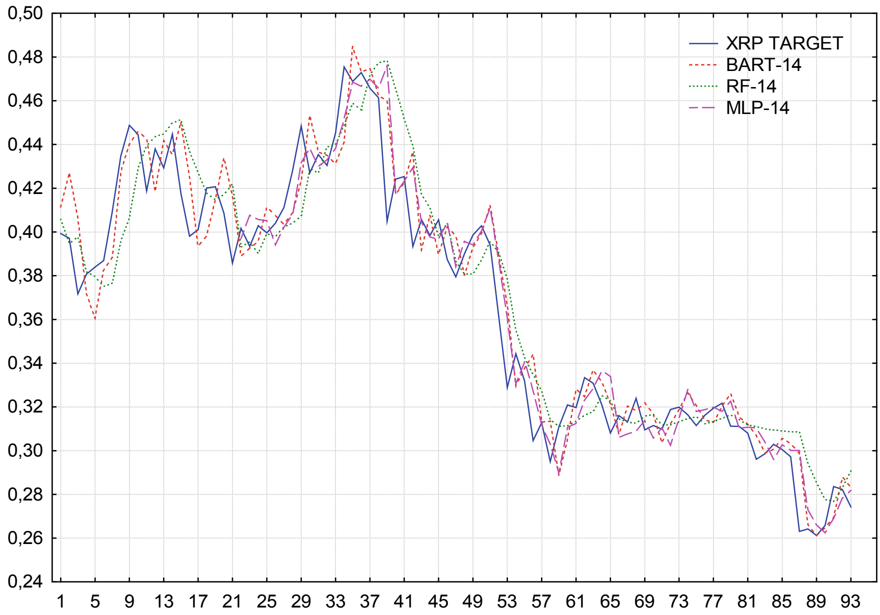

Fig. 8 90-day forecast of XRP prices (lag $p = 14$)

the 90-day time horizon, which was carried out using one-step forecasting technique without adjusting models parameters.

Analysis of the graphs allows us to conclude that the models fit the real data sufficiently well, taking into account the complex oscillating dynamic behavior of the studied series: an increasing trend for BTC and a decreasing one for ETH and XRP.

We can also observe that all models, despite the overall adequacy of the existing trends in the cryptocurrency dynamics, show some delay relative to the real data.

For estimating prediction accuracy we used metrics of Mean Percentage Absolute Error (MAPE) and Root Mean Square Error (RMSE):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|Y_i - \hat{f}(Y_i)\right|}{Y_i} \times 100\%, \quad RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{f}(Y_i)\right)^2}. \quad (15)$$

It should be noted that RMSE can only be used to evaluate the quality of different forecasts for one financial assets (time series). It provides information about the magnitude of the error. But RMSE does not characterize this error in comparison to the actual quote value.

In contrast, MAPE allows evaluating the forecasts performance of both individual models and their ensembles for different assets and compare them with each other.

Table 1 Out-of-sample accuracy performance results for different lags

	BTC		ETH		XRP	
	MAPE, %	RMSE	MAPE, %	RMSE	MAPE, %	RMSE
Lag $p = 7$						
BART-7	3.71	535.2	**3.39**	**11.74**	**3.07**	**0.0154**
RF-7	7.11	971.9	7.44	21.8	3.94	0.0196
MLP-7	**3.69**	**529.8**	3.53	12.17	**3.07**	**0.0153**
Lag $p = 14$						
BART-14	**3.83**	**541.9**	**3.37**	**11.86**	3.42	0.0167
RF-14	5.60	756.9	6.48	19.82	4.08	0.0203
MLP-14	3.95	559.1	3.51	12.16	**3.41**	**0.0162**
Lag $p = 21$						
BART-21	**3.94**	**558.5**	**3.69**	**12.55**	3.83	0.0183
RF-21	5.54	739.3	4.52	14.55	3.92	0.0212
MLP-21	4.28	610.8	3.84	13.17	**2.98**	**0.0151**

In our evaluation of predictive accuracy, we made a forecast of the dynamics of cryptocurrency prices over a 90-day horizon by using one-step forecasting technique.

The final out-of-sample accuracy results obtained from the BART, MLP and RF are shown in Table 1.

The accuracy obtained from both BART and MLP are significantly higher for all lags and cryptocurrencies than for the RF algorithm. The relatively low accuracy of RF may be due to the fact that a much larger number of factors are required for its effective implementation. It is worth noting that RF accuracy increases as the depth of the lag increases. Accuracy can also be improved by building more trees in the forest.

As for the comparison of the MLP and BART performance, the results in Figs. 6, 7 and 8 and Table 1 show similar accuracy of these models: the smallest error (MAPE) for BTC was 3.69% (MLP), for ETH—3.37% (BART), for XRP—2.98% (MLP).

Somewhat unexpected, there was a slight decrease in the accuracy of both the MLP and BART (at least for BTC and ETH) with increasing lag depth. In our opinion, this may be due to the overfitting problem.

From the trading point of view it is more valuable to predict the direction of price or trend change, rather than its numerical value. Since all three types of models can solve the classification problem we also performed prediction of the price change direction of BTC, ETH and XRP from August 1, 2019 to December 1, 2019 (123 observations).

To investigate this problem, we made forecast for growth (class positive, P) and falling (class negative, N) prices on the next day by using one-step forecasting technique without adjusting the model parameters.

However, a certain observation was classified as positive, P or negative, N if the price of the asset for that day increased (or decreased) by 1% or more, respectively.

Table 2 Prediction accuracy of the prices change direction of individual cryptocurrencies for the period 01/08/19–01/12/19

		Actual	BART		MLP		RF	
			Pred.	Accur. %	Pred.	Accur. %	Pred.	Accur. %
BTC	Rising, *P*	29	21	**64**	17	**62**	19	**57**
	Falling, *N*	53	31		28		29	
ETH	Rising, *P*	41	29	**62**	25	**59**	27	**59**
	Falling, *N*	49	27		24		26	
XRP	Rising, *P*	49	33	**59**	29	**61**	31	**56**
	Falling, *N*	52	26		25		25	

To measure forecasting performance, we used Accuracy metrics defined in (16) below, which represents the proportion of correctly predicted values among all predictions

$$Accuracy = \frac{TP + TN}{P + N}, \tag{16}$$

were TP and TN are the number of correctly predicted values of positive and negative classes, respectively; P and N are the actual number of values for each class. Table 2 shows the summary of the estimation accuracy of our models by using this metric.

As shown in Table 2, the prediction accuracy of the BART and MLP are higher for all time series than for the RF models. The average values of the Accuracy metric by the BART model are 62%, MLP 61%, RF 57%.

Note that for all models the proportion of correctly predicted values of the positive class (increase in price), turned out to be higher than the proportion of the correctly predicted values of the decrease in price, which must be taken into account in practical application of the models.

We can conclude based on the considered accuracy metrics defined in (15-16), that the models of the short-term forecast of the cryptocurrency prices dynamics in general have smaller errors than the “naive forecasts”. During the periods of slow change, these models can be used to make a short-term forecasts for up to 30 days.

For traders with a longer investment horizon (90 days to a year) it is necessary to take into account the dynamics of nonlinear trends, and in our opinion, it would be advisable to use the models developed by us in combination with trend-cycles models.

6 Conclusion

The results of our modeling of short-term cryptocurrency dynamics and application of these models to real life data demonstrated the effectiveness of using machine learning approach, in particular, models of neural networks, regression (autoregressive) trees and their ensembles for forecasting tasks. Based on the results of the study, these models allow making short-term forecast with sufficient accuracy: within 3–4%.

Results of the binary classification of the direction of price changes showed, that BART and MLP models had an average accuracy of about 63% for the daily time series observations, which was higher than for the "naive" model.

It should be noted that we used a minimal dataset—only lag values of the studied series (closing prices). Forecast accuracy can be increased by using a more expanded dataset: including open, maximum, minimum and average prices, trading volume, etc. In addition, we can use a variety of indices, oscillators, in particular, moving averages of different types and time periods, taking into account the trend dynamics.

In this work we have applied a simple model of Neural Network—the Multilayer Perceptron with one hidden layer. Using networks with more complex architecture: recurrent, self-organized, deep, etc. should also improve the predictive accuracy. In summary, we note that the perspective approach for the financial time series forecasting is the construction of combined Classification and Regression Tree models and Neural Networks.

References

Akyildirim, E., Goncuy, A., & Sensoy, A. (2018). *Prediction of Cryptocurrency Returns Using Machine Learning*. https://www.researchgate.net/publication/329322600. Accessed November 15, 2019.

Albuquerque, Y., de Sá, J., Padula, A., & Montenegro, M. (2018). The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression. *Expert Systems with Applications, 97,* 177–192. https://doi.org/10.1016/j.eswa.2017.12.004.

Alessandretti, L., ElBahrawy, A., Aiello, L., & Baronchelli, A. (2018). Anticipating cryptocurrency prices using machine learning. *Hindawi Complexity*. https://doi.org/10.1155/2018/8983590.

Amjad, M., & Shah, D. (2016). *Trading Bitcoin and online time series prediction*. NIPS 2016 Time Series Workshop. http://proceedings.mlr.press/v55/amjad16.pdf. Accessed November 15, 2019.

Belinskyi, A., Soloviev, V., Semerikov, S., & Solovieva, V. (2019). Detecting stock crashes using Levy distribution. In *CEUR Workshop Proceedings* (Vol. 2422, pp. 420–433). http://ceur-ws.org/Vol-2422/paper_34.pdf.

Bontempi, G., Taieb, S., & Borgne, Y. (2013). Machine learning strategies for time series forecasting. In *European Business Intelligence Summer School eBISS 2012* (pp. 62–77). Berlin, Heidelberg: Springer.

Boyacioglu, M., & Baykan, O. K. (2011). 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, 38*(5), 5311–5319.

Breiman, L. (2001). Random forests. *Machine Learning, 45,* 5–32.

Breiman, L., Friedman, H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, NJ: Wadsworth International Group.

Catania, L., & Grassi, S. (2017). *Modelling crypto-currencies financial time-series*. CEIS Research Paper (15(8), pp. 1–39). https://ideas.repec.org/p/rtv/ceisrp/417.html. Accessed November 15, 2019.

Cheah, E. (2015). Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of bitcoin. *Economic Letters, 130,* 32–36.

Ciaian, P. (2016). The economics of BitCoin price formation. *Applied Economics, 48*(19), 1799–1815.

CNBC. (2018). *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 November 15, 2019.

Derbentsev, V., Datsenko, N., Stepanenko, O., & Bezkorovainyi, V. (2019a). Forecasting cryptocurrency prices time series using machine learning. In *CEUR Workshop Proceedings* (Vol. 2422, pp. 320–334).

Derbentsev, V., Kibalnyk, L., & Radzihovska, Y. (2019b). Modelling multifractal properties of cryptocurrency market using Hurst exponent and detrended fluctuation analysis. *PEN, 7*(2), 690–701.

Flach, P. (2012). *Machine learning: The art and science of algorithms that make sense of data*. Cambridge, UK: Cambridge University Press.

Hitam, N. A., & Ismail, A. R. (2018). *Comparative Performance of Machine Learning Algorithms for Cryptocurrency Forecasting*. https://www.researchgate.net/publication/327415267. Accessed November 15, 2019.

Kennis, M. (2018). *A Multi-channel Online Discourse as an Indicator for Bitcoin Price and Volume*. arXiv:1811.03146v1 [q-fin.ST]. Accessed November 6, 2018.

Krugman, P. (2013). *Bits and Barbarism*. http://www.nytimes.com/2013/12/23/opinion/krugmanbits-and-barbarism.html. Accessed November 15, 2019.

Kumar, M. (2006). *Forecasting stock index movement: A comparison of support vector machines and random forest*. SSRN Working Paper. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=876544. Accessed November 15, 2019.

Matviychuk, A. (2006). Fuzzy logic approach to identification and forecasting of financial time series using Elliott wave theory. *Fuzzy Economic Review, 11*(2), 51–68.

Matviychuk, A. V. (2011). Shtuchnyi intelekt v ekonomitsi: neironni merezhi, nechitka logika (Artificial Intelligence in Economics: Neural Networks, Fuzzy Logic), Kyiv, KNEU (in Ukrainian).

McNally, S. (2016). *Predicting the price of Bitcoin using machine learning* (Doctoral dissertation). National College of Ireland, Dublin.

Okasha, M. K. (2014). Using support vector machines in financial time series forecasting. *Statistics, 4*(1), 28–39. https://doi.org/10.5923/j.statistics.20140401.03.

Peng, Y., Henrique, P., & Albuquerque, M. (2018). The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression. *Expert Systems with Applications, 97,* 177–192.

Persio, L., & Honchar, O. (2018). Multitask machine learning for financial forecasting. *International Journal of Circuits, Systems and Signal Processing, 12*, 444–451.

Popper, N. (2015). *Digital gold: Bitcoin and the inside story of the misfits and millionaires trying to reinvent money*. New York, NY: Harper Collins Publisher.

Rebane, J., & Karlsson, I. (2018). Seq2Seq RNNs and ARIMA models for cryptocurrency prediction: A comparative study. In *SIGKDD Fintech'18*. https://doi.org/10.475/123_4.

Saxena, A., & Sukumar, T. (2018). Predicting bitcoin price using LSTM and compare its predictability with ARIMA model. *International Journal of Pure Applied Mathematics, 119*(17), 2591–2600.

Selmi, R., Tiwari, A., & Hammoudeh, S. (2018). Efficiency or speculation? A dynamic analysis of the Bitcoin market. *Economic Bulletin, 38*(4), 2037–2046.

Soloviev, V., & Belinskij, A. (2016). *Methods of Nonlinear Dynamics and the Construction of Cryptocurrency Crisis Phenomena Precursors*. arXiv:1807.05837; https://arxiv.org/abs/1807.05837. Accessed November 15, 2019.

Soloviev, V., & Belinskij, A. (2019). Complex systems theory and crashes of cryptocurrency market. In *Communications in Computer and Information Science* (Vol. 1007, pp. 276–297). https://link.springer.com/chapter/10.1007/978-3-030-13929-2_14.

Soloviev, V., Belinskij, A., & Solovieva, V. (2019a). Entropy analysis of crisis phenomena for DJIA index. In *CEUR Workshop Proceedings* (Vol. 2393, pp. 434–449). http://ceur-ws.org/Vol-2393/paper_375.pdf.

Soloviev, V., Serdiuk, O., Semerikov, S., & Kohut-Ferens, O. (2019b). Recurrence entropy and financial crashes. In *Proceedings of the 7th International Conference on Modeling, Development and Strategic Management of Economic System*, Ivano-Frankivsk, Ukraine, October 24–25, 2019. https://www.atlantis-press.com/proceedings/.

Soloviev, V., Solovieva, V., Tuliakova, A., & Ivanova, M. (2019c). Construction of crisis precursors in multiplex networks. In *Proceedings of the 7th International Conference on Modeling, Development and Strategic Management of Economic System*, Ivano-Frankivsk, Ukraine, October 24–25, 2019. https://www.atlantis-press.com/proceedings/.

Varghade, P., & Patel, R. (2012). Comparison of SVR and decision trees for financial series prediction. *IJACTE, 1*(1), 101–105.

Vigna, P., & Casey, M. J. (2015). *The age of cryptocurrency: How Bitcoin and digital money are challenging the global economic order*. New York, NY: St. Martin's Press.

Wang, M., Zhao, L., Du, R., Wang, C., Chen, L., Tian, L., et al. (2018). A novel hybrid method of forecasting crude oil prices using complex network science and artificial intelligence algorithms. *Applied Energy, 220,* 480–495. https://doi.org/10.1016/j.apenergy.2018.03.148.

Yahoo Finance. (2019). https://finance.yahoo.com. Accessed November 15, 2019.

Yao, Y., Yi, J., & Zhai, S. (2018). Predictive analysis of cryptocurrency price using deep learning. *International Journal of Engineering & Technology, 7*(3.27), 258–264.