International Journal of Engineering

Journal Homepage: www.ije.ir

Time Series Forecasting of Bitcoin Price Based on Autoregressive Integrated Moving Average and Machine Learning Approaches

M. Khedmati*[a], F. Seifi[a], M. J. Azizi[b]

[a] *Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran*
[b] *Daniel J. Epstein department of industrial and systems engineering, University of Southern California, Los Angeles, United States*

PAPER INFO

Paper history:
Received 19 January 2020
Received in revised form 13 March 2020
Accepted 11 June 2020

Keywords:
Time Series Forecasting
Machine Learning
Bitcoin
Multivariate Models

ABSTRACT

Bitcoin as the current leader in cryptocurrencies is a new asset class receiving significant attention in the financial and investment community and presents an interesting time series prediction problem. In this paper, some forecasting models based on classical like ARIMA and machine learning approaches including Kriging, Artificial Neural Network (ANN), Bayesian method, Support Vector Machine (SVM) and Random Forest (RF) are proposed and analyzed for modelling and forecasting the Bitcoin price. While some of the proposed models are univariate, the other models are multivariate and as a result, the maximum, minimum and the opening daily price of Bitcoin are also used in these models. The proposed models are applied on the Bitcoin price from December 18, 2019 to March 1, 2020 and their performances are compared in terms of the performance measures of RMSE and MAPE by Diebold-Mariano statistical test. Based on RMSE and MAPE measures, the results show that SVM provides the best performance among all the models. In addition, ARIMA and Bayesian approaches outperform other univariate models where they provide smaller values for RMSE and MAPE.

***doi**: 10.5829/ije.2020.33.07a.16*

1. INTRODUCTION

Time series forecast plays an important role in many fields such as economics, finance, business intelligence, meteorology, and telecommunication [1]. As such, time series forecasting has been an active area of research since 1950s and many empirical and theoretical studies are conducted [2-4]. As an early attempt, researchers tried to use linear combination of historical data and hence, most of the traditional statistical models including moving average, exponential smoothing, and autoregressive integrated moving average (ARIMA) have linear structure [5]. However, in the late 1970s, it became increasingly clear that linear models, per se, are not adapted for many applications like stochastic series [1]. Therefore, nonlinear models like autoregressive conditional heteroscedastic (ARCH) and general autoregressive conditional heteroscedastic (GARCH) were introduced. In last two decades, Machine Learning (ML) models have established themselves as serious rivals of classical models in forecasting literature [6-9]. ML models are examples of potentially nonparametric and nonlinear models which use only historical data to learn the stochastic dependency between the historical date and future [1].

There is a growing interest on financial time series forecasting in recent years because it plays a significant role in investment decisions. Generally, financial time series have noise characteristic due to the unavailability of complete information while their non-stationary characteristic originates in the distributional changes over time. In other words, financial time series forecasting is a relatively challenging task [10].

Recently, Cryptocurrencies (i.e. digital monetary systems stored in an encrypted block-chain) have received significant attention in the financial community [11]. The current supposed leader of Cryptocurrencies, Bitcoin, presents an interesting time series rising in a market that is in its transient stage [12]. In this paper, we propose some forecasting models based on ARIMA and ML methods to forecast the price of Bitcoin. The proposed ML approaches include Kriging, Artificial

Corresponding Author Email: khedmati@sharif.edu* (M. Khedmati)

Please cite this article as: M. Khedmati, F. Seifi, M.J. Azizi, Time Series Forecasting of Bitcoin Price Based on Autoregressive Integrated Moving Average and Machine Learning Approaches, International Journal of Engineering (IJE), IJE TRANSACTIONS A: Basics Vol. 33, No. 7, (July 2020) 1293-1303

Neural Networks (ANNs), Bayesian model, Support Vector Machines (SVMs), and Random Forest (RF). Moreover, we use the opening, maximum, and minimum daily price in addition to the closing price to improve the prediction.

The rest of the paper is organized as follows. Section 2 provides a literature review on Bitcoin price forecasting studies. The ARIMA approach is explored in detail in Section 3. In Section 4, the proposed ML models are studied, and their performance are compared in Section 5. Finally, the concluding remarks constitute Section 6.

2. LITERATURE REVIEW

The studies in this literature are divided into two categories; one uses the Bitcoin features to predict its price while the other one has an economic point-of-view. This paper focuses on the latter. This category consists of two subcategories including the *classical methods* of transformations and the *Machine Learning* models on data like opening, maximum, minimum and gold prices.

2. 1. Classical Approach In this category, Chu et al. [13] applied statistical analysis on the exchange rate log-returns of Bitcoin versus the US Dollar. This paper compared 15 popular financial parametric distributions on the log returns and concluded that the generalized hyperbolic distribution provides the best results. The financial capabilities of Bitcoin are studied in some papers including Dyhrberg [14]. They showed several similarities of Bitcoin to gold and dollar, indicating hedging capabilities and advantages of Bitcoin as a medium of exchange. Autoregressive approaches are also studied in which for example, Hencic and Gouriéroux [15] used the mixed causal-noncausal autoregressive process with Cauchy errors to predict the Bitcoin price. In addition, Ho et al. [16] compared ARIMA, recurrent and multilayer feed-forward networks showing that the first two outperform the last model. Some of the studies use transformations where, for example, Delfin-Vidal and Romero-Melendez [17] used a continuous wavelet transform analysis on the price volatility across different time and investment horizons.

In another spectrum of studies, the authors used Bitcoin attributes and economical tools. For example, Kristoufek [18] addressed the price changes focusing on possible sources of the change, ranging from fundamentals to speculative or technical sources. This work examined how interconnections behave in time with different scales (frequencies). In another study, Kristoufek [19] used a similar approach and studied the relationship between digital currencies, such as Bitcoin, and Google Trends or Wikipedia search queries. This study showed not only that they are connected, but also there exists a pronounced asymmetry between the effect of an increased interest in the currency when it is above or below its trending value. Another work that looked into these types of interconnections is Garcia et al. [20] that used the data from social media and search engines. They studied the links between social signals and Bitcoin prices through a social feedback cycle and found two main positive feedback loops indicating a strong connection.

2. 2. Machine Learning Approaches The Bayesian and linear regression variants have been used extensively to predict the Bitcoin price. For instance, Shah and Zhang [21] employed Bayesian regression as the latent source model and devised a simple strategy for trading Bitcoin. Another example is Greaves and Au [22] which applied linear regression, logistic regression, SVM, and ANN on the block-chain network-based features of the price. In a slightly altered approach, Madan et al. [23] proposed two phases; first, they used over 25 characteristics of the price and payment network over 5 years to predict the sign of future changes using Binomial General Linear Model (GLM), SVM, and RF models. Afterward, for the second phase, they merely focused on the Bitcoin price data, alone.

Deep learning approaches have also been employed for Bitcoin price predictions. Almeida et al. [24] focused on the prediction of the price trend for the next day based on the previous days' price and volume using an ANN model. McNally et al. [12] showed that nonlinear deep learning models including Bayesian optimized Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network can outperform ARIMA. A number of research efforts including Sin and Wang [25], Radityo et al. [26], Indera et al. [11], Jang and Lee [27] and Rahimi and Khashei [28] focused on applying specialized deep learning models. Reinforcement learning models also can be helpful due to the feedback interacting behavior of the price forecast problem. For instance, Lee et al. [29] proposed to predict the price movements using Inverse Reinforcement Learning (IRL) and Agent-Based Modelling (ABM). Their model reproduces synthetic yet realistic rational agents in a simulated market.

As we mentioned in some instances, utilizing a social network or search engine data is effective for modelling and forecasting where, this is shown in Matta et al. [30]. They investigated the relation between the spread of the Bitcoin price and volumes of tweets or Web Search media results, particularly those with a positive sentiment. They explored significant cross-correlations, especially on the Google Trends data.

Sentimental analysis on Twitter feeds can reveal fundamental economic variables and technological factors. Georgoula et al. [31] used this fact to study the relationship between Bitcoin prices and the information derived from the tweets.

3. DATA DESCRIPTION

In this paper, we use the daily Bitcoin exchange rate data (the closed price of Bitcoin) from December 18, 2019, to March 1, 2020, from [32]. The closing prices are the common target of prediction in the literature. Also, note that since the data of this period has a one-time growth, based on the suggestion of economists, we do not use that data. A statistical summary of these data is presented in Table 1. In this table, Mean, SD, Min, and Max represent the mean, standard deviation, minimum, and maximum. In addition, the data is plotted in Figure 1, which obviously suggests non-stationarity of the process.

In the multivariate models, in addition to the Bitcoin price, the maximum, minimum and opening daily prices are also used. Here, we use 75 observations (days) for training the models and the last 44 records as test dataset for one-step-ahead prediction.

4. METHODOLOGY

In this paper, the ARIMA model as the classic method and ANN, SVM, RF, Bayesian method, and Kriging as the machine learning models are chosen for forecasting of Bitcoin price. Then, we analyze the models, compare the results of them and select the best model based on the performance measures.

4. 1. Classic Method ARIMA models developed by Box and Jenkins [33] have been widely used for time series forecasting. An ARIMA model is usually linear, combined by several previous observations and random errors, and the prediction model is created as a function of the historical data and the errors [34]. The conventional ARIMA (p, d, q) formulation is described as:

$$\Phi(B)(1-B)^d y_t = \delta + \Theta(B)\varepsilon_t \quad (1)$$

in which δ is a constant term, $\Phi(B)$ is the autoregressive coefficient function, $\Theta(B)$ is the moving average coefficient function, ε_t is the error term at time *t* and *d* is the order of integration terms. If the time series is stationary, then *d* is zero and the model simplifies to ARMA (Autoregressive Moving Average). We first examine the stationarity of the data using Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [35]. In this test, the null and the alternative hypothesis are as below:

Hypothesis 0. y_t is a unit root process.

Hypothesis 1. y_t is trend (or level) stationary or $\sigma_u^2 = 0$

where,

- $y_t = \beta_t + (r_t + \kappa) + e_t$ and y_t are observations during the time
- $r_t = r_{t-1} + u_t$ is a random walk process with $r_0 = \kappa$
- *t* represents time
- $u_t \sim NID(0, \sigma_u^2)$

The value of the KPSS test statistic for Bitcoin data (resp. P-value) is 1.7376 (resp. 0.01), and therefore the null hypothesis cannot be rejected with a 95% confidence level. Now, the sources of non-stationarity should be identified. This can be due to the existence of a trend in the data or changes in their variance. To test the former hypothesis (trend) on the data, we use the Mann-Kendall test [36-38] because of its relevance to the matter. In this test, the null and the alternative hypotheses are as below:

Hypothesis 0. No monotonic trend.

Hypothesis 1. The monotonic trend is present.

The P-value of this test for Bitcoin data is 1.697e-9. Hence, we reject the null hypothesis and use differencing on the data to remove the trend. After differencing, the Mann-Kendall test is used again for the integrated data where, the related P-value is changed to 0.4553. This P-value represents the lack of trend in the transformed data. Figure 2 shows the time series data after differencing.

Now, we must check the variance stability for transformed data. To do this, the Breusch-Pagan [39] test is used based on the following hypotheses.

Hypothesis 0. Variance is homoscedastic.

Hypothesis 1. Variance is not homoscedastic.

The P-value of this test for Bitcoin data is 2.2e-16, which does not lead to the rejection of the null hypothesis and hence, we have a stable variance. In addition, the P-value of KPSS test for the transformed data is 0.1. Therefore, we cannot reject the null hypothesis of KPSS test and accordingly, the transformed data is stationary.

Once the data is stationary, it is time to select the model in which the Extended Sample Auto Correlation

TABLE 1. Statistical summary of the daily Bitcoin exchange rate data, N = 75

Variable	Mean	SD	Min	Max
Close price	8660.36	1022.66	6967.00	10333.00
Open price	8634.43	1049.73	6613.50	10336.00
High price	8807.38	1038.49	7197.60	10482.60
Low price	8478.33	1007.36	6462.20	10229.30

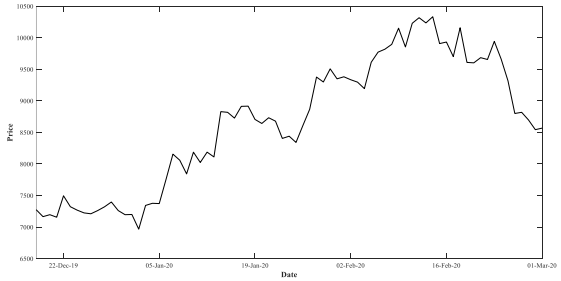

Figure 1. Bitcoin price from December 18, 2019 to March 1, 2020

Function (ESACF) is used for this purpose. Based on ESACF matrix, an ARMA (1,1) is appropriate for the transformed data. Note that since differencing is used to convert the data into a stationary time series, the resulting model is ARIMA (1,1,1) and its parameters for forecasting the price on March 1, 2020, are:

$$(1-0.6034B)(1-B)y_t = -31.27 + (1-0.4086B)\varepsilon_t \qquad (2)$$

Figure 3 shows the residual plots of this model and Figure 4 shows the ACF (Autocorrelation Function) while PACF (Partial Autocorrelation Function) plots. Since there is no specific trend in these diagrams, it can be assumed that the model is properly selected and can be used for prediction.

There are standard measures for evaluating the performance of forecasting models where, Woschnagg and Cipan [40] address some of these methods. In this paper, we use RMSE (Root Mean Square Error) and MAPE (Mean Absolute Percentage Error) to evaluate the accuracy of the proposed models. This is because RMSE is a beneficial measure for comparing the accuracy of the models and MAPE is relatively easy to interpret. Specifically, the formulas are:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}$$
$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|(\hat{y}_t - y_t)\right| \qquad (3)$$

Following an online schema, we update our model parameters after removing the oldest point and adding the new one. The RMSE and MAPE for the proposed ARIMA model are 253.4513 and 2.2286%, respectively.

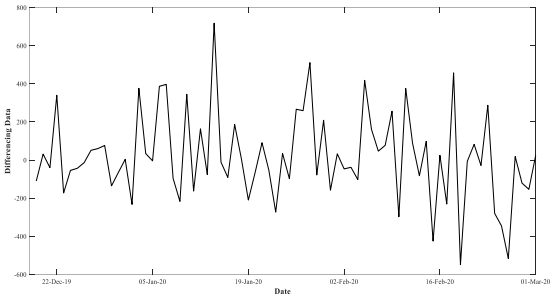

Figure 2. Bitcoin price with first order differencing

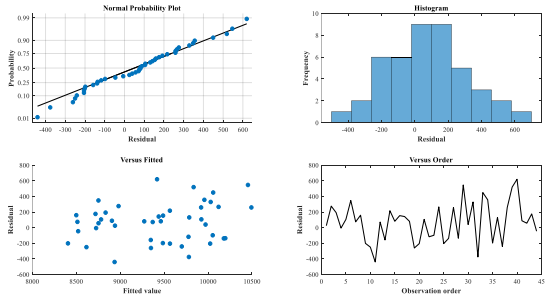

Figure 3. Residual plots

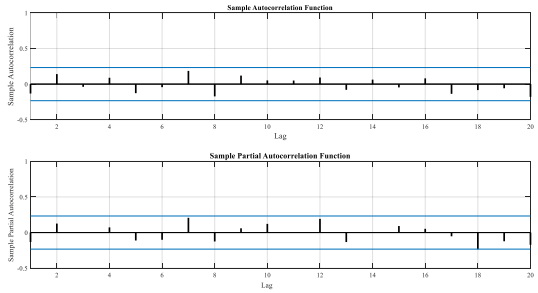

Figure 4. The ACF (top) and PACF (down) plots of residuals

4. 2. Machine Learning Methods For the proposed multivariate ML models, the variables of opening, maximum, and minimum daily prices are used in addition to the closing prices, while the univariate model just uses the closing prices. The univariate models include Kriging, ANN, and Bayesian methods and multivariate models include ANN, SVM, RF, and Bayesian methods. This section ends with the comparison of the models with each other.

4. 2. 1. Kriging Kriging is the interpolation of unknown values in a stochastic function with a linear weighted set of observed values [41]. Krige, an African mining engineer, invented this method to define the exact location of mining rocks in the 1950s [42]. The main idea of this meta-model is to use a weighted mean of outputs in such a way that the weights depend on the interspace between forecasting point and observed points. The optimal weights give minimum prediction error variance and the predictions are the Best Likelihood Unbiased Estimators (BLUE). Due to these properties, Kriging is an optimal interpolator [43], i.e. Kriging meta-models traverse through all the members of the experimental environment. This model is mainly used for prediction purposes in addition to sensitivity analysis and robust optimization. Generally, Kriging is classified into six categories, namely Simple, Ordinary, Co-Kriging, Universal, Blind and Stochastic [43]. To the best of our knowledge, there are just a few studies that use Kriging for time series forecasting. For example, Cellura et al. [44] applied a neural Kriging method to the spatial estimation of wind speed for energy planning in Sicily and, Liu et al. [45] used Kriging for prediction of wind speed.

In this paper, we use a univariate ordinary Kriging (OK) model with a simple parameter tuning, which indicated that using the last 5 days' data provides the best performance. Ordinary Kriging interpolates the one-step-ahead-forecast (y_n+1) using a set of n existing records (y_i; i =1, …, n). We suppose that the mean output is the unknown variable and the prediction is expressed as follows:

$$y_{n+1} = \lambda + \gamma \Gamma^{-1}\left(\boldsymbol{y}^T - \mathbf{1}\lambda\right) \quad (4)$$

where, $\lambda = \left(\mathbf{1}^T \Gamma^{-1}\mathbf{1}\right)^{-1}\mathbf{1}^T\Gamma^{-1}\boldsymbol{y}^T$, $\mathbf{1}$ is a $n\times 1$ vector of ones, $\Gamma = cov\left(y_i, y_{i'}\right); i = 1,\ldots,n, i' = 1,\ldots,n$ a $n\times n$ matrix of covariance between the data points, $\gamma = cov\left(y_i, y_{n+1}\right)$ a $n\times 1$ vector including covariance between the data y_i and the prediction y_{n+1}, and finally, $\boldsymbol{y}$, a $n\times 1$ vector of the Bitcoin prices. The prediction variance is calculated as:

$$S^2_{y_{n+1}} = \sigma^2\left(1 - \gamma\Gamma^{-1}\gamma^T + \frac{1-\mathbf{1}^T\Gamma^{-1}\gamma^T}{\mathbf{1}^T\Gamma^{-1}\mathbf{1}}\right) \quad (5)$$

where, $\sigma^2 = \frac{1}{n}\left(\boldsymbol{y}^T - \mathbf{1}\lambda\right)^T\Gamma^{-1}\left(\boldsymbol{y}^T - \mathbf{1}\lambda\right)$ [46]. We use ooDACE toolbox of MATLAB as in [47]. The proposed model cares less about the past data compared to the new data, and only the data from the past 5 days is used to fit the model. In this regard, the Kriging model is very similar to a moving average model but the covariance matrix helps to obtain the parameters. The RMSE and MAPE of this model are 298.1863 and 2.6768%, respectively.

4. 2. 2. Bayesian Method The basic theory of prediction using Gaussian processes goes back to Wiener [48] and Kolmogorov [49] in the 1940s. Indeed, Lauritzen [50] discusses the relevant work by Danish astronomer T. N. Thiele from 1880 [51]. The Bayes rule is one of those simple but profound ideas that underlie statistical thinking [52]. However, finding an appropriate prior distribution for the data is a difficult task, where it makes the Bayesian analysis more complicated than other models to utilize.

As a result of the central limit theorem, the Gaussian processes are flexible enough to have a good performance as a prior distribution on many data sets with a large number of data [53]. The goal of the Bayesian forecasting is to compute the distribution P($y_{n+1}|D$, n+1) of output y_{n+1} given a test input $n+1$ and a set of n training records D ={(i, y_i)| i =1, …, n}. We use the Bayes rule to obtain the posterior distribution for the ($n+1$)th Gaussian process outputs. The prediction conditioned on the observed outputs has Gaussian distribution [54]; that is:

$$P(y_{n+1} \mid D, n+1) \sim N\left(\mu_{y_{n+1}}, \sigma^2_{y_{n+1}}\right) \quad (6)$$

where, the mean and variance are given by $\mu_{y_{n+1}} = \gamma^T\Gamma\boldsymbol{y}^T$ and $\sigma^2_{y_{n+1}} = cov\left(y_{n+1}, y_{n+1}\right) - \gamma^T\Gamma\gamma$, respectively. Since the variance is predicted for the ($n+1$)th Gaussian process output, the confidence interval of the prediction, in addition to the point prediction, can be obtained [53]. We use both univariate and multivariate Bayes models and the results in Table 2 show that the multivariate model has better accuracy than the univariate one.

4. 2. 3. Artificial Neural Network Artificial neural networks (ANN) can recognize future patterns of the time series. ANNs are universal and very flexible function approximators, first used in the fields of cognitive science and engineering. One of the similar works on ANN in the financial studies is Kaastra and Boyd [55], which provides an eight-step procedure to design an ANN forecasting model. They also discuss the trade-offs in parameter selection, some common pitfalls, and points of disagreement among practitioners. In addition, Azoff [56] takes the reader of their book beyond the 'black-box' approach to neural networks and provides the knowledge that is required for their proper design and use in financial markets forecasting with an emphasis on futures trading. Zhang [57] proposed a hybrid methodology that combines both ARIMA and ANN models to benefit the strength of ARIMA and ANN models in the linear and nonlinear modeling.

The feed-forward neural networks are used widely in the literature [58-59], where, we propose a feed-forward neural network with a single hidden layer and lagged inputs to forecast univariate time series. The historical data is the input of the neural network model, while the output is the forecast value. The hidden layer stores an appropriate transfer function which is used for processing the data from the input nodes. The model is expressed as:

$$y_{n+1} = w_0 + \sum_{j=1}^{Q} w_j g(w_{0j} + \sum_{i=0}^{P} w_{i,j} y_{n-i}) \quad (7)$$

where, P is the number of input nodes, Q the number of hidden nodes, g an activation function, {w_j, j = 0, 1, …, Q} a vector of weights from the hidden layer to output nodes, {$w_{i,j}$, i =1, 2, …, P, j = 0, 1, …, Q} are the weights between the input to hidden nodes and $w_{0,j}$ are the weights for each output between input and hidden layer [34]. To select the number of previous observations, features that should be in the model (feature selection) and the number of the nodes in the hidden layer, we use parameter tuning

TABLE 2. RMSE and MAPE for Bayesian model

Model	RMSE	MAPE (%)
Univariate	245.9793	2.0551
Multivariate	192.8742	1.5251

TABLE 3. RMSE and MAPE for ANN model

Model	RMSE	MAPE (%)
Univariate	321.2871	2.8218
Multivariate	252.0465	2.0865

on the training data set with R language. The results of univariate and multivariate ANN models are represented in Table 3.

4. 2. 4. Support Vector Machine The motivation for using the support vector machines (SVMs) in time series forecasting is the ability of this methodology to accurately forecast time series data when the underlying processes are typically nonlinear, non-stationary and not defined a-priori. This model is also shown to outperform other non-linear techniques including neural-network-based non-linear prediction techniques such as multi-layer perceptrons [60].

The general idea of SVM for regression (or SVR) is to generate the regression function by applying a set of high dimensional linear functions. Then, it uses a minimization on the upper bound of the generalization error [34]. The inputs are mapped into a high dimensional nonlinearly feature space (F), wherein the features are correlated linearly with the outputs. The SVR formulation considers the following linear estimation function [10]:

$$f(\boldsymbol{t}) = \boldsymbol{w}^T \phi(\boldsymbol{t}) + \boldsymbol{b} \tag{8}$$

where $\boldsymbol{w}$ is the weight vector, $\boldsymbol{b}$ the bias term vector, $\phi(\boldsymbol{t})$ denotes a mapping function in the feature space and $\boldsymbol{w}^T\phi(\boldsymbol{t})$ the dot production in the feature space F. Various cost functions such as the Laplacian, Huber's Gaussian and Vapnik's linear ε-Insensitivity can be used in the SVR formulation. Among these, the Vapnik's linear ε-Insensitivity loss function is the most commonly adopted [38], which is given in Equation (9).

$$\left|y_i - f(t_i)\right|_\varepsilon = \begin{cases} 0 & if \left|y_i - f(t_i)\right| \le \varepsilon \\ \left|y_i - f(t_i)\right| - \varepsilon & otherwise \end{cases} \tag{9}$$

where, ε is a precision parameter representing the radius of the tube located around the regression function $f(t)$. Accordingly, linear regression $f(t)$ is estimated by simultaneously minimizing $\|\boldsymbol{w}\|^2$ and the sum of the linear ε-Insensitivity losses as bellow:

$$R = \frac{1}{2}\|\boldsymbol{w}\|^2 + c\left(\sum_{i=1}^{n}\left|y_i - f(t_i)\right|_\varepsilon\right) \tag{10}$$

in which the constant c controls the weight of approximation error and size of weights vector, $\|\boldsymbol{w}\|$. Increasing c, potentially decreases the approximation error with a trade-off that controls the overfitting. Minimizing the risk R is equivalent to the model given in Equations (11a-11d) [5].

$$\text{minimize } R = \frac{1}{2}\|\boldsymbol{w}\|^2 + c\left(\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right)\right) \tag{11-a}$$

subject to

$$\left(\boldsymbol{w}^T t_i + b\right) - y_i \le \varepsilon + \xi_i \quad \forall i = 1,\ldots,n \tag{11-b}$$

$$y_i - \left(\boldsymbol{w}^T t_i + b\right) \le \varepsilon + \xi_i^* \quad \forall i = 1,\ldots,n \tag{11-c}$$

$$\xi_i, \xi_i^* \ge 0 \quad \forall i = 1,\ldots,n \tag{11-d}$$

By using the Lagrangian multipliers and Karush-Kuhn-Tucker conditions, the following dual Lagrangian model is obtained as [61]:

$$\text{maximize } L\left(\alpha, \alpha^*\right) \tag{12-a}$$

$$\text{subject to } \sum_{i=1}^{n}\left(\alpha_i^* - \alpha_i\right) = 0 \tag{12-b}$$

$$0 \le \alpha_i \le C \quad \forall i = 1,\ldots,n \tag{12-c}$$

$$0 \le \alpha_i^* \le C \quad \forall i = 1,\ldots,n \tag{12-d}$$

with the following definitions:

$$\begin{aligned} L\left(\alpha,\alpha^*\right) &= -\varepsilon\sum_{i=1}^{n}\left(\alpha_i^* + \alpha_i\right) + \sum_{i=1}^{n}\left(\alpha_i^* - \alpha_i\right)y_i \\ &-\frac{1}{2}\sum_{i,j=1}^{n}\left(\alpha_i^* - \alpha_i\right)\left(\alpha_j^* - \alpha_j\right)K\left(t_i, t_j\right) \end{aligned} \tag{13}$$

Note that the Lagrangian multipliers in Equation (13) satisfy the equality $\alpha_i\alpha_i^* = 0$ and the optimal weights for the regression is $\left(\boldsymbol{w}^T\right)^* = \sum_{i=1}^{n}\left(\alpha_i^* - \alpha_i\right)K(\boldsymbol{t}, t_i)$. Hence, the general form of the SVR-based regression function can be written, according to Vapnik [61] as follows:

$$f(\boldsymbol{t}) = \sum_{i=1}^{n}\left(\alpha_i^* - \alpha_i\right)K(\boldsymbol{t}, t_i) + \boldsymbol{b} \tag{14}$$

where, $K(\boldsymbol{t}, t_i)$ is the kernel function which is proportional to the inner product of two points, t_i and t_j, in the feature space $\phi(t_i)$ and $\phi\left(t_j\right)$; that is, $K\left(t_i, t_j\right) = \phi(t_i)\phi\left(t_j\right)$. Although several choices for the kernel function are available, the most widely used is the Radial Basis Function (RBF) $K(t_i, t_j) = \exp(-\|t_i - t_j\|^2/2v^2)$, where, v denotes the width of the RBF [10]. We use the "SVM" package and "tune" function of R for parameter tuning. The results of univariate and multivariate SVM models are represented in Table 4.

4. 2. 5. Random Forest Random Forest (RF) is an ensemble learning method for classification and regression that proceeds by constructing an aggregation of decision trees [62]. Random decision forests adjust for

the decision trees' habit of overfitting to their training set. In the context of time series, the changes in the future data are dynamic and a regression might not be an excellent choice however, we can tune it at the current time of a sliding window. An example of a random forest application in time series is the work in Kane et al. [54]. They showed that using random forest enhances the predictive ability over existing time series models for the prediction of infectious disease outbreaks in bird populations. Using RF for one-step-ahead time series forecasting is straightforward and similar to the application of RF in the regression models. Let f be the model function which will be used for y_{n+1}, given y_1, …, y_n. If we use k lagged variables, the predicted y_{n+1} is obtained based on the following equation for $t = n+1$ [63]:

$$y_t = f\left(y_{t-1},\ldots,y_{t-k}\right), t = k+1,\ldots,n+1 \quad (15)$$

We use a training set of size n-k. In each training sample, the dependent variable is y_t, for $t = k+1$, …, $n+1$, while the predictor variables are y_{t-1}, …, y_{t-k}. When k increases, the size of the training set n-k decreases. The training set, which includes n-k samples, is created using the "CasesSeries" function of the "rminer" R package [64-65]. The RMSE and MAPE of this method are obtained as 237.0867 and 2.0787%, respectively.

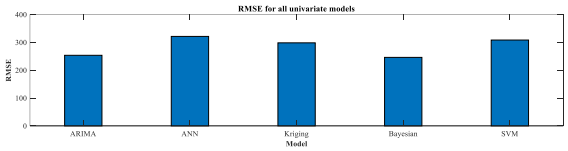

Figure 5. RMSE for all univariate models

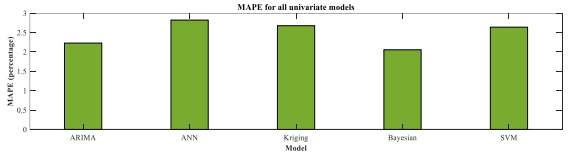

Figure 6. MAPE for all univariate models

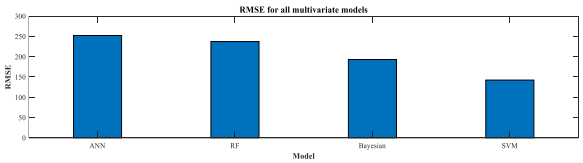

Figure 7. RMSE for all multivariate models

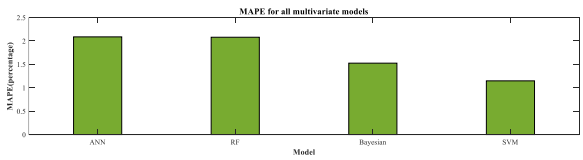

Figure 8. MAPE for all multivariate models

TABLE 4. RMSE and MAPE for SVM model

Model	RMSE	MAPE (%)
Univariate	308.2759	2.6397
Multivariate	142.1580	1.1469

5. PERFORMANCE COMPARISON

In this section, the models are compared based on the results of the previous section. The results of univariate models are shown in Figures 5 and 6. In this regard, considering the univariate models, Bayesian has shown the best performance with the smallest values of RMSE and MAPE among other models. In addition, the results of the multivariate models are shown in Figures 7 and 8. The results show that among the multivariate models, SVM has fantastic performance and its accuracy is significantly better than the other multivariate models. Furthermore, we plot the cumulative RMSE for all the univariate and multivariate models in Figures 9 and 10, respectively. Based on the results, in the univariate case, the Bayesian and ARIMA models and in the multivariate case, the SVM model outperform other models consistently through time.

A more systematic way to compare two time series models is to look at their errors in pairs where the Diebold-Mariano test is used for this purpose. Let y_t as the actual value for time point t and $\hat{y}_{it}$ be the prediction of the model i for time t. Then, the error for model i in time t is defined as $e_{it} = \hat{y}_{it} - y_t$. Now, the loss differential between the forecast of two models is defined as $d_t = g(e_{1t}) - g(e_{2t})$ where, $g(e_{it})$ shows the loss function. In this regard, we have [66]:

$$\sqrt{T}(\bar{d} - \mu) \rightarrow N(0, 2\pi f_d(0)) \quad (16)$$

where, $\bar{d} = \sum_{t=1}^{T} d_t / T$ and $f_d(0) = \sum_{\tau=-\infty}^{\infty} \gamma_d(\tau) / 2\pi$ is the spectral density of the loss differential at frequency zero. In addition,

$$\gamma_d(\tau) = E[(d_t - \mu)(d_{t-\tau} - \mu)] \quad (17)$$

is the autocovariance of the loss differential at lag τ. Under the null hypothesis that the two forecasts have the same accuracy, the Diebold-Mariano test statistic is defined as:

$$DM = \bar{d} / \sqrt{\left(2\pi \hat{f}_d(0)/T\right)} \quad (18)$$

in which, $\hat{f}_d(0) = \frac{1}{2\pi} \sum_{k=-(T-1)}^{(T-1)} I(\frac{k}{h-1}) \hat{\gamma}_d(k)$ is the estimate of $f_d(0)$ with:

$$\hat{\gamma}_d(k)=\frac{1}{T}\sum_{t=|k|+1}^{T}(d_t-\bar{d})(d_{t-|k|}-\bar{d}) \quad (19)$$

and $I(x)=1$ if $|x|\le 1$ and 0, otherwise. In practice, if we set $M=T^{1/3}$, an appropriate estimate of $2\pi f_d(0)$ is obtained by $\sum_{k=-M}^{M}\hat{\gamma}_d(k)$, and hence we have:

$$DM=\bar{d}\Big/\sqrt{\left(\sum_{k=-M}^{M}\hat{\gamma}_d(k)/T\right)} \quad (20)$$

Under the null hypothesis, the test statistics DM is asymptotically $N(0,1)$ distributed and the null hypothesis is rejected if $|DM|>Z_{\alpha/2}$.

We use this test for the univariate and multivariate models in which the pairwise P-value of Diebold-Mariano statistic for univariate and multivariate time series are represented in Tables 5 and 6, respectively. In this paper, we applied one sided null hypothesis; that is, the value at row *i* and column *j* is the P-value testing if model *i* is better than model *j*. For example, in Table 5, 0.0283 is the P-value of the hypothesis which indicates that the performance of the ARIMA model is better than the Kriging model.

We use a 95% confidence level and follow a round robin policy. The results show that the ARIMA and Bayesian models have the same accuracy and outperform other univariate models. It should be noted that despite the better performance of the ARIMA model, it is harder to fit because of its complex data pre-processing including the differencing and various data transformations. However, machine learning models are faster than the ARIMA model with relatively high accuracy. For example, the accuracy of the Bayesian model is the same as the accuracy of ARIMA, while it provides the results much faster than the latter. In other words, although machine learning models need hyper-parameter tuning, this is easier than the pre-processing step of the ARIMA model. In addition, the effect of hyper-parameter tuning on the performance of ML models is less than the effect of pre-processing on the performance of the ARIMA model.

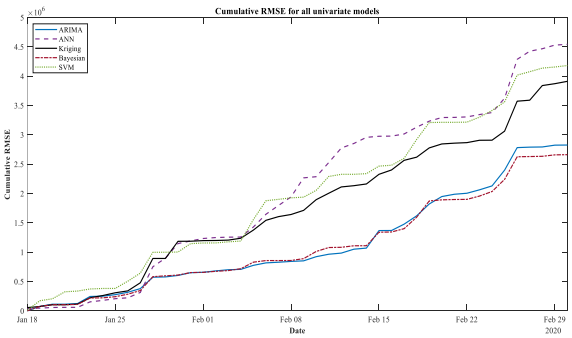

Figure 9. Cumulative RMSE for univariate models

Based on the results in Table 6, the SVM model has the best accuracy between multivariate models which confirms the comparison based on RMSE and MAPE criteria. Hence, we identify the SVM model as the best multivariate model. The outputs of the best univariate and multivariate models are shown in Figures 11 and 12.

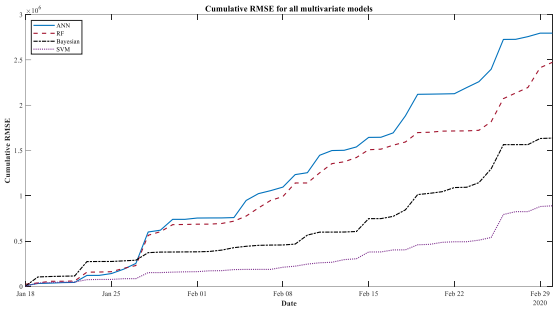

Figure 10. Cumulative RMSE for multivariate models

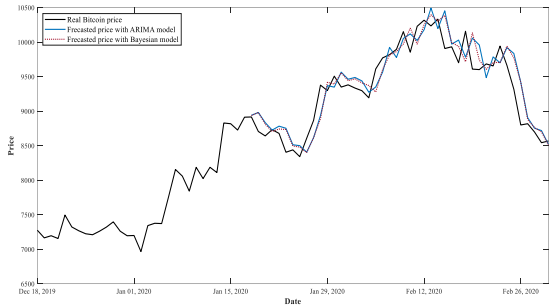

Figure 11. Real Bitcoin price with forecasted them form best univariate models

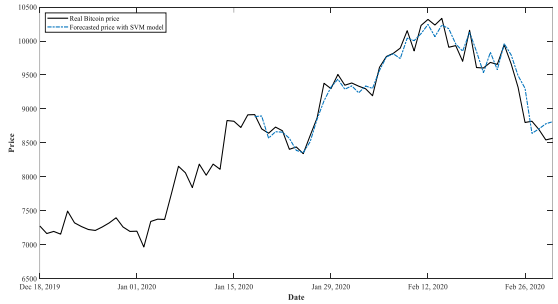

Figure 12. Real Bitcoin price with forecasted them form SVM model

TABLE 5. P-value of DM test for all univariate models

	ARIMA	Kriging	Bayesian	SVM	ANN
ARIMA	-	0.0283	0.7793	0.0158	0.0137
Kriging	0.9716	-	0.9881	0.3418	0.1403
Bayesian	0.2207	0.0119	-	0.0020	0.0064
SVM	0.9841	0.6582	0.9980	-	0.3195
ANN	0.9862	0.8597	0.9935	0.6805	-

TABLE 6. P-value of DM test for all multivariate models

	Bayesian	RF	SVM	ANN
Bayesian	-	0.0294	0.998	0.0062
RF	0.9706	-	1	0.1921
SVM	0.0020	4e-05	-	8e-05
ANN	0.9841	0.6582	0.998	-

6. CONCLUDING REMARKS

In this paper, the time series forecasting approaches of ARIMA as a classical model and five machine learning models including Kriging, artificial neural network (ANN), Bayesian method, support vector machine (SVM) and random forest (RF) are proposed for modeling and forecasting of Bitcoin price. The proposed models included the univariate and multivariate models in which the ARIMA, ANN, Kriging and Bayesian models are used as univariate while ANN, SVM, RF, and Bayesian models are proposed for multivariate case. Then, these models are applied on the Bitcoin price from December 18, 2019, to March 1, 2020, where, in addition to the Bitcoin price, the maximum, minimum and opening price of Bitcoin for the same period is also used in the multivariate models. Comparing the performance of the proposed models in terms of the RMSE and MAPE measurements, it is concluded that ARIMA and Bayesian provide better results compared to other univariate models since they have smaller RMSE and MAPE values compared to other models. However, the SVM outperforms all the univariate and multivariate models and is selected as the best model where its performance measures of RMSE and MAPE are much smaller than the values of all other models.

As a recommendation for future research, one can consider the effect of variables other than the ones related to the Bitcoin price in the multivariate models for possible improvement in the forecasting. Another avenue for future research can be designing a way for hyper-parameter optimization of the investigated models.

7. REFERENCES

1. Bontempi, G., Ben Taieb, S., and Le Borgne, Y. A., Machine Learning Strategies for Time Series Forecasting. ***European Business Intelligence Summer School***, Vol. 138, (2012), 62-77. doi: 10.1007/978-3-642-36318-4_3
2. Stock, J. H. and Watson, M. W., A Comparison of Linear and Nonlinear Univariate Models for Forecasting Macroeconomic Time Series. ***National Bureau of Economic Research***, Vol. 14, No. 1, (1996), 11-30. doi: 10.3386/w6607
3. Lemke, C. and Gabrys, B., Meta-learning for time series forecasting and forecast combination. ***Neurocomputing***, *73*(10-12), (2010), 2006-2016. doi: 10.1016/j.neucom.2009.09.020
4. Ahmadia, E., Abooiea, M. H., Jasemib, M., and Zare Mehrjardi, Y., A Nonlinear Autoregressive Model with Exogenous Variables Neural Network for Stock Market Timing: The Candlestick Technical Analysis. ***International Journal of Engineering. Transactions C: Aspects***, Vol. 29, No. 12, (2016), 1717-1725. doi: 10.5829/idosi.ije.2016.29.12c.10
5. Patel, J., Shah, S., Thakkar, P., and Kotecha, K., Predicting stock market index using fusion of machine learning techniques. ***Expert Systems with Applications***, Vol. 42, No. 4, (2015), 2162-2172. doi: 10.1016/j.eswa.2014.10.031
6. Ahmed, N. K., Atiya, A. F., Gayar, N. E., and 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
7. Palit, A. K. and Popovic, D., *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications.* Springer, (2005).
8. Neshat, N., An Approach of Artificial Neural Networks Modeling Based on Fuzzy Regression for Forecasting Purposes. ***International Journal of Engineering. Transactions B: Applications***, Vol. 28, No. 11, (2015), 1651-1655. doi: 10.5829/idosi.ije.2015.28.11b.13
9. Werbos, P. J., Generalization of backpropagation with application to a recurrent gas market model. ***Neural Networks***, Vol. 1, No. 4, (1988), 339-356. doi: 10.1016/0893-6080(88)90007-X
10. Lu, C., Lee, T., and Chiu, C., Financial time series forecasting using independent component analysis and support vector regression. ***Decision Support Systems***, Vol. 47, No. 2, (2009), 115-125. doi: 10.1016/j.dss.2009.02.001
11. Indera, N. I., Yassin, I. M., Zabidi, A., and Rizman, Z. I., Non-linear autoregressive with exogeneous input (narx) Bitcoin price prediction model using pso-optimized parameters and moving average technical indicators. ***Journal of Fundamental and Applied Sciences***, Vol. 9, No. 3, (2017), 791-808. doi: 10.4314/jfas.v9i3s.61
12. McNally, S., Roche, J., and Caton, S., Predicting the Price of Bitcoin Using Machine Learning. 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), (2018). doi: 10.1109/PDP2018.2018.00060
13. Chu, J., Nadarajah, S., and Chan, S., Statistical Analysis of the Exchange Rate of Bitcoin. ***PLoS ONE***, Vol. 10, No. 7, (2015), doi: 10.3386/t0090
14. Dyhrberg, A. H., Bitcoin, gold and the dollar – A GARCH volatility analysis. ***Finance Research Letters***, Vol. 16, (2016), 85-92. doi: 10.1016/j.frl.2015.10.008
15. Hencic, A. and Gouriéroux, C. .Noncausal Autoregressive Model in Application to Bitcoin/USD Exchange Rates. ***Econometrics of Risk***, Vol. 583, (2014), 17-40. doi: 10.1007/978-3-319-13449-9_2
16. Ho, S. L., Xie, M., Goh, T. N., A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction. ***Computers & Industrial Engineering***, Vol. 42, (2002), 371-375. doi: 10.1016/S0360-8352(02)00036-0
17. Delfin-Vidal, R. and Romero-Meléndez, G., The Fractal Nature of Bitcoin: Evidence from Wavelet Power Spectra. ***Trends in Mathematical Economics***, (2016), 73-98. doi: 10.1007/978-3-319-32543-9_5
18. Kristoufek, L., What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis. ***PLoS ONE***, Vol. 4, No. 10, (2015). doi: 10.1371/journal.pone.0123923
19. Kristoufek, L., Bitcoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era. ***Scientific Reports***, Vol. *3*, (2013), 1-7, doi: 10.1038/srep03415.

20. Garcia, D., Tessone, C. J., Mavrodiev, P., and Perony, N., The digital traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy. ***Journal of the Royal Society Interface***, Vol. 11, No. 99, (2014), 1-8. doi: 10.1098/rsif.2014.0623
21. Shah, D. and Zhang, K., Bayesian regression and Bitcoin. 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), (2014), 409-414. doi: 10.1109/ALLERTON.2014.7028484
22. Greaves, A. and Au, B., Using the Bitcoin transaction graph to predict the price of Bitcoin, (2015). http://snap.stanford.edu/class/cs224w-2015/projects_2015/.
23. Madan, I., Saluja, S., and Zhao, A., Automated Bitcoin trading via machine learning algorithms, (2015). http://cs229.stanford.edu/proj2014/.
24. Almeida, J., Tata, S., Moser, A., and Smit, V. Bitcoin prediciton using ANN. ***Neural Networks***, Vol. 7, (2015), 1-12.
25. Sin, E. and Wang, L., Bitcoin price prediction using ensembles of neural networks. 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), (2017), 666-671. doi: 10.1109/FSKD.2017.8393351
26. Radityo, A., Munajat, Q., and Budi, I., Prediction of Bitcoin exchange rate to American dollar using artificial neural network methods. International Conference on Advanced Computer Science and Information Systems (ICACSIS), (2017), 433-438. doi: 10.1109/ICACSIS.2017.8355070
27. Jang, H. and Lee, J., An Empirical study on modeling and prediction of Bitcoin prices with Bayesian neural networks based on blockchain information. ***IEEE Access***, Vol. 6, (2017), 5427-5437. doi: 10.1109/ACCESS.2017.2779181
28. Rahimi, Z. H., Khashei, M., A least squares-based parallel hybridization of statistical and intelligent models for time series forecasting. ***Computers & Industrial Engineering***, (2018), doi: https://doi.org/10.1016/j.cie.2018.02.023.
29. Lee, K., Ulkuatam, S., Beling, P., and Scherer, W. . Generating synthetic Bitcoin transactions and predicting market price movement via inverse reinforcement learning and agent-based modeling. ***Journal of Artificial Societies and Social Simulation***, Vol. 21, No. 3, (2018), 1-5. doi: 10.18564/jasss.3733
30. Matta, M., Lunesu, I., and Marchesi, M., Bitcoin Spread Prediction Using Social and Web Search Media. UMAP Workshops, (2015).
31. Georgoula, I., Pournarakis, D., Bilanakos, C., Sotiropoulos, D., and Giaglis, G. M., Using Time-Series and Sentiment Analysis to Detect the Determinants of Bitcoin Prices. ***MCIS 2015 Proceedings***, Vol. 20, (2015). doi: 10.2139/ssrn.2607167
32. www.investing.com/crypto/bitcoin/historical-data. (n.d.).
33. Box, P. and Jenkins, G., Time series analysis: forecasting and control. Wiley, (1976).
34. Shi, J., Guo, J., and Zheng, S. . Evaluation of hybrid forecasting approaches for wind speed and power generation time series. ***Renewable and Sustainable Energy Reviews***, Vol. 16, No. 5, (2012), 3471-3480. doi: 10.1016/j.rser.2012.02.044
35. Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., and Shin, Y., Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? ***Journal of Econometrics***, Vol. 54, No. 1-3, (1992), 159-178.
36. Mann, H. B. , Non-parametric tests against trend. ***Econometrics***, Vol. 13, (1945), 163-171. doi: 10.2307/1907187
37. Gilbert, R. O., Statistical methods for environmental pollution monitoring. John Wiley & Sons, (1987).
38. Kendall, M. G., *Rank correlation methods*, 4th Edition, Charles Griffin, London, (1975).
39. Breusch, T. S. and Pagan, A. R., A simple test for heteroskedasticity and random coefficient variation. ***Econometrica***, Vol. 47, No. 5, (1979), 1287-1294. doi: 10.2307/1911963
40. Woschnagg, E. and Cipan, J., Evaluating forecast accuracy. UK Ökonometrische prognose, department of economics, university of Vienna, (2004). http://homepage.univie.ac.at/robert.kunst/procip.pdf.
41. Ankenman, B., Nelson, B. L., and Staum, J., Stochastic Kriging for simulation metamodeling. ***Operations research***, Vol. 58, No. 2, (2010), 371-382. doi: 10.1109/WSC.2008.4736089
42. Kleijnen, J. P., Kriging metamodeling in simulation: A review. ***European Journal of Operational Research***, Vol. 192, No. 3, (2009), 707-716. doi: 10.1016/j.ejor.2007.10.013
43. Journel, A. G., Fundamentals of geostatistics in five lessons. *American Geophysical Union, Washington, D.C.* , (1989).
44. Cellura, M., Cirrincione, G., Marvuglia, A., and Miraoui, A., Wind speed spatial estimation for energy planning in Sicily: A neural Kriging application. ***Renewable Energy***, Vol. 33, No. 6, (2008), 1251-1266. doi: 10.1016/j.renene.2007.08.012
45. Liu, H., Shi, J., and Erdem, E., Prediction of wind speed time series using modified Taylor Kriging method. ***Energy***, Vol. 35, No. 12, (2010), 4870-4879. doi: 10.1016/j.energy.2010.09.001
46. Azizi, M. J., Seifi, F., Moghadam, S., A robust simulation optimization algorithm using Kriging and particle swarm optimization: Application to surgery room optimization. ***Communications in Statistics-Simulation and Computation***, (2019), doi: 10.1080/03610918.2019.1593452
47. Couckuyt, I. Dhaene, T. Demeester, P., ooDACE Toolbox: A Flexible Object-Oriented Kriging Implementation. ***Journal of Machine Learning Research***, Vol. 15, (2014), 3183-3186.
48. Kolmogorov, A., Interpolation und Extrapolation von stationären zufälligen Folgen. ***Izv. Akad. Nauk SSSR Ser. Mat.***, Vol. 5, No. 1, (1941), 3-14.
49. Lauritzen, S. L., Time Series Analysis in 1880: A Discussion of Contributions Made by T.N. Thiele. ***International Statistical Institute (ISI)***, Vol. 49, No. 3, (1981), 319-331. doi: 10.2307/1402616
50. Rasmussen, C. E. and Williams, C. K. I., Gaussian Processes *in Machine Learning*. MIT press, (2006).
51. Efron, B., *Large-Scale Inference Empirical Bayes Methods for Estimation, Testing, and Prediction.* Vol. 1, Cambridge University Press, (2012).
52. Brahim-Belhouari, S., and Bermak, A., Gaussian process for nonstationary time series prediction. ***Computational Statistics & Data Analysis***, Vol. 47, No. 4, (2004), 705-712. doi: 10.1016/j.csda.2004.02.006
53. Kane, M. J., Price, N., Scotch, M., and Rabinowitz, P. . Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks. BMC Bioinformatics, (2014), Vol. 15:276, doi: 10.1186/1471-2105-15-276
54. Kaastra, I. and Boyd, M., Designing a neural network for forecasting financial and economic time series. ***Neurocomputing***, Vol. 10, No. 3, (1996), 215-236.
55. Azoff, E. M., *Neural Network Time Series Forecasting of Financial Markets.* New York: John Wiley & Sons, (1994).
56. Zhang, G. P., Time series forecasting using a hybrid ARIMA and neural network model. ***Neurocomputing***, Vol. 50, (2003), 159-175. doi: 10.1016/S0925-2312(01)00702-0
57. Khashei, M. and Bijari, M., An artificial neural network (p, d, q) model for time series forecasting. ***Expert Systems with Applications***, Vol. 37, No. 1, (2010), 479-489. doi: 10.1016/j.eswa.2009.05.044

58. Zhang, G. P. and Qi, M., Neural network forecasting for seasonal and trend time series. ***European Journal of Operational Research***, Vol. 160, No. 2, (2005), 501-514. doi: 10.1016/j.ejor.2003.08.037

59. Sapankevych, N. I., Sankar, R., Time Series Prediction Using Support Vector Machines: A Survey. ***IEEE Computational Intelligence Magazine***, Vol. 4, No. 2, (2009), 24-38. doi: 10.1109/MCI.2009.932254

60. Vapnik, V. N., The Nature of Statistical Learning Theory. Springer, (2000).

61. Ho, T. K., Random decision forests. *In Proceedings of 3rd International Conference on Document Analysis and Recognition*, Vol. 1, (1995), 278-282. doi: 10.1109/ICDAR.1995.598994

62. Tyralis, H. and Papacharalampous, G. . Variable Selection in Time Series Forecasting Using Random Forests. ***Algorithms***, Vol. 10, No. 4, (2017), 114. doi: 10.3390/a10040114

63. Liaw, A. and Wiener, M., Classification and regression by random forest. ***R News***, Vol. 2, No. 3, (2002), 18-22.

64. Cortez, P., Data Mining with Neural Networks and Support Vector Machines Using the R/rminer Tool. ***Advances in Data Mining. Applications and Theoretical Aspects***, (2010), 572-583. doi: 10.1007/978-3-642-14400-4_44

65. Diebold, F. X., Comparing Predictive Accuracy. ***Journal of Business & Economic Statistics***, Vol. 13, No. 3, (1995), 253-263.

Persian Abstract

چکیده

بیت‌کوین به عنوان رمزارز پیشرو، دسته‌ی جدیدی از دارائی‌هاست که در فضای مالی و سرمایه‌گذاری توجه زیادی را به خود جلب کرده است و در نتیجه یک مسئله جذاب پیش‌بینی سری زمانی را ارائه می‌کند. در این مقاله، برخی از روش‌های پیش‌بینی کلاسیک نظیر ARIMA در کنار روش‌های یادگیری ماشین شامل کریگینگ، شبکه‌ی عصبی مصنوعی، روش بیزین، ماشین بردار پشتیبان و جنگل تصادفی جهت مدل‌سازی و پیش‌بینی ارزش بیت‌کوین به کار برده شده‌اند. در این مقاله، برخی از مدل‌ها به صورت تک متغیره و برخی دیگر به صورت چندمتغیره با بهره‌گیری از بیشترین و کمترین ارزش روز و قیمت بازگشایی بازار بیت‌کوین برازش شده‌اند. مدل‌های ارائه شده، بر روی ارزش بیت‌کوین در بازه‌ی زمانی ۱۸ دسامبر ۲۰۱۹ تا ۱ مارس ۲۰۲۰ اعمال شده و عمل‌کرد آنها بر اساس معیارهای ریشه‌ی میانگین مربعات خطا و میانگین قدرمطلق درصد خطا در کنار آزمون آماری Diebold-Mariano با یک‌دیگر مقایسه شده است. برمبنای معیارهای ریشه‌ی میانگین مربعات خطا و میانگین قدرمطلق درصد خطا، نتایج نشان می‌دهد که ماشین بردار پشتیبان دارای بهترین عملکرد در بین تمامی مدل‌هاست. همچنین، مدل‌های ARIMA و بیزین با در اختیار داشتن مقادیر پایین ریشه‌ی میانگین مربعات خطا و میانگین قدرمطلق درصد خطا، بهترین عمل‌کرد را در بین مدل‌های تک‌متغیره دارا هستند.