Applied Soft Computing 129 (2022) 109584

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

MLP-based Learnable Window Size for Bitcoin price prediction

Shahab Rajabi [a], Pardis Roozkhosh [b], Nasser Motahari Farimani [c,*]

[a] Department of Electrical Engineering, K. N. Toosi University of Technology, Tehran 19967-15433, Iran
[b] Department of Management, Faculty of Economics and Administrative Sciences, Ferdowsi University of Mashhad, Mashhad 9177948951, Iran
[c] Department of Management, Ferdowsi University of Mashhad, Mashhad 9177948951, Iran

ARTICLE INFO

Article history:
Received 13 November 2021
Received in revised form 8 June 2022
Accepted 17 August 2022
Available online 30 August 2022

Keywords:
Bitcoin
Deep neural network
LWS
PHF
Blockchain

ABSTRACT

Over the past few years, Bitcoin price prediction has been changed to a big challenge for investors on cryptocurrencies. In this regard, Neural Networks as a strong structure for regression analysis would play an important role to make a precise prediction. While several leading researches in this field considered the features affecting the price of bitcoin by a fixed number of past days, a new method entitled Learnable Window Size (LWS) is presented for smartening the number of days intended to predict the price of Bitcoin the next day. This paper implements a primary deep neural network, based on the observed Bitcoin price trend in the past days and its fluctuations, to predict the best window size. Then, the secondary deep neural network predicts the price of Bitcoin according to the predicted window of the first step. The dataset of this paper is included Google, Blockchain, and Bitcoin market data. Evaluations have shown that based on the Prediction Hardship Factor (PHF), a new criterion which has been proposed to describe the degree of difficulty of prediction, this method has been able to get the minimum error under a normal situation which is superior in comparison to the well-known methods such as Support Vector Regression and ARIMA.

1. Introduction

Nowadays, due to the globalization of the economy and its prevalence, we are moving from traditional methods of trading and investing to new ones. As a result, digital currency has significantly helped us to encourage paperless transactions. Transaction mediation and security are the biggest issues that every investor has to accept. In every aspect of business, human is combining technology and seeking to reduce the day-to-day business. Marketing information systems, enterprise resource planning, and advanced software such as Systems Applications and Products (SAP) are used in accounting to improve its performance [1]. Additionally, because digital currencies are not controlled by the central bank, they allow transactions to be made without any intermediaries. However, market forces determine the price of digital currency [2,3]. To better understand the benefits of using digital currency, a revision was provided for a taxonomy of centralization in decentralized Blockchains under harsh synthesis [4]. This new type of currency has the characteristics of goods and money at the same time and is named "Cryptocurrency" [5]. In this regard, the well-known cryptocurrency, "Bitcoin", was introduced in order to function as a payment system. Recent investigations are about the various effects of Bitcoin on the environment and its inefficient market, fluctuations, and Nature [6,7]. On this hand, a new survey was showed the importance of cryptocurrency and suggested that if users preferred Bitcoin instead of the official currency, what would happen to governments that disagree with Bitcoin. Although some authorities have even taken steps to prohibit their citizens from trading in Bitcoin, other nations, such as New Zealand, have allowed Bitcoin payments [8].

One of the most important aspects of using cryptocurrencies, both as an intermediary and as an asset, is the expected amount prediction. Predicting the price of Bitcoin, as a popular digital currency, has been a major part of academic work [9]. One of the key points of digital currencies that have attracted more attention is whether such markets are in line with the Efficient Markets Hypothesis (EMH) [1]. Besides, a lot of research and methods have been done to predict the price of digital currencies. For example, artificial neural networks are widely used to predict financial markets by using technical indicators [10–15]. Plus, market sentiment indices were used as stock market predictors [16–19]. Furthermore, the Estimation Maximum Likelihood (EML) was used to show the Bitcoin market is completely Efficient [20]. Nevertheless, making accurate predictions in a complex and rapid analytical framework is still certainly a challenging issue.

In recent years, various studies have been done on bitcoin price prediction such as Arima-based and SVM-based regression models using time series data and kernel function respectively. Random forest also is a supervised learning algorithm that uses

* Corresponding author.
E-mail addresses: sh.rajabi@ee.kntu.ac.ir (S. Rajabi), pardis.roozkhosh@mail.um.ac.ir (P. Roozkhosh), n.motahari@um.ac.ir (N.M. Farimani).

https://doi.org/10.1016/j.asoc.2022.109584
1568-4946/

multiple machine learning method for forecasting. The majority of the papers in this field considered the features affecting the price of Bitcoin using an MLP neural network or a Long Short-Term Memory (LSTM) structure. Jana et al. (2021) studied the bitcoin daily price behavior for forecasting. They used support vector regression (SVR) and polynomial regression with interaction (PRI) [21]. Dhinakaran et al. (2022) used the traditional Autoregressive Integrative Moving Average (ARIMA) for analyzed the bitcoin daily price for 3 years. They showed this method can have a fit forecast for bitcoin price [22]. Nikol et al. (2021) examined the bitcoin price hourly and daily from 2017 to 2019. They used the random forest (RF) method and showed that this method can improve the errors. In their study, the random forest had better results in comparison with ARIMA [23].

In this paper, as a novel idea, two separate MLP-based neural networks have been combined to create a more robust prediction system. In this regard, the first MLP-based deep neural network is used to smarten the number of sequential days (window size) for Bitcoin price prediction. The process is such that a primary neural network based on price fluctuations over a period of one to seven days determines what is the best window size to include the effective past days for the next day Bitcoin price. Then, a secondary deep neural network, trained based on the predicted window size, is used to predict the price of bitcoin.

The dataset used in this paper includes 21 different features of market factors, Blockchain, and Google testimonials. In order to improve the prediction operation in the second neural network, in addition to the values of these features, the daily fluctuations of each of them are also considered as a supportive feature. Rest of the paper is organized as follows. Section 2 investigates recent methods that have been done to predict Bitcoin price. Then, the proposed method in this paper is explained in detail and at the end, the statistical results and the robustness of the proposed method are illustrated.

2. Background

Myriad of features are considered to affect the cryptocurrency price prediction performance such as trading volume and Hash rate [24,25]. Bitcoin is presented as the first unconcerned, non-seasonal impact and encrypted digital currency of the online market which influences the International economy significantly [26]. Bariviera (2017) studied daily Bitcoin data to check its fluctuations. The results of this study show that there is no confidence in the bitcoin market, but over time it takes steps towards stability [27]. Furthermore, Alvarez-Ramirez et al. (2018) by analyzing fluctuations of Bitcoin price, found that the Bitcoin market contains a period of time, in which price is dynamic, and there were asymmetric correlations by increase and decreasing prices [28]. One of the most important structures aiming for Bitcoin price prediction is AI-based methods. In recent years, the interest in long-term analysis of Bitcoin behavior has been dramatically increased. Jay et al. (2020) represented a neural network according to casual walk theory, which is used to predict stock exchange prices in financial markets [19]. Accordingly, Berat et al. (2020) provided a completed revision from accomplishing deep learning (DL) studies for Bitcoin price prediction, and it is found that DL-based Regional News Network (RNN) models are the best structures [18,29,30]. Besides, Koo and Kim (2021) studied Multi-Layer Perceptron (MLP), RNN including Long Short-Term Memory (LSTM) to evaluate the performance of flattening distribution strategies [31]. In another method, economic and technological factors were applied as features to predict the price of Bitcoin using the LSTM structure [32]. Additionally, Awoke et al. (2021) provided a new method using DL models to predict the fluctuations of Bitcoin price with LSTM and Gated Recurrent Unit (GRU) structure [33]. One-dimensional Convolutional Neural Network(CNN) is a strong method that would create a strategy for Bitcoin price prediction which has been illustrated that the results have been improved compared to the LSTM structure in some specific situations [34]. Recently, by studying the trend of Bitcoin price during the COVID-19 pandemic, it is observed that fluctuations are very high and the stock market affects Bitcoin price owing to the period's high uncertainty [35]. Choi and Kshin (2021) obtained the same result with the investigation about the effect of inflation on Bitcoin price [36]. In general, there is a significant relationship between the COVID-19 period, Bitcoin price, and social media that would be perceived in the short and long term before this pandemic [37,38]. Also,

Most of the methods used in previous studies to predict the price of Bitcoin are generally based on deep neural networks. In the following, recent advanced methods such as ARIMA, RF, SVR, WaveNet, and LSTM that have been implemented in this field are briefly explained.

2.1. Regression

Regression models describe the relationship between the dependent variable y and the independent variables x. The general model of multiple linear regression is in Eq. (1).

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ip} + \varepsilon_i \quad i = 1, \ldots, n. \tag{1}$$

Where n is the total number of observations, y_i is the output corresponding to the $\boldsymbol{i}^{th}$ observation, p is the total number of independent variables, x_{ij} is the number of $\boldsymbol{i}^{th}$ observation corresponding to the $\boldsymbol{j}^{th}$ of the independent variable, β_k values represent the weight of each variable, β_0 is the constant value of the model, and ε_i is $\boldsymbol{i}^{th}$ independent distributed error. Overall, a regression model with a fitted linear function would be modeled as Eq. (2).

$$y_i = \beta_0 + \sum_k \beta_{ik} f_k \left(x_{ij}\right) + \varepsilon_i. \tag{2}$$

Where f is a function with the output of a scalar value of the independent variables x_{ij} [39]. The Box–Jenkins methodology refers to a set of procedures for identifying and estimating of time series models within the class of ARIMA models [40]. ARIMA models are a class of models that have capabilities to represent stationary time series and to produce accurate predictions based on a description of historical data of a single variable. The ARIMA (p,d,q) model has three parts corresponding to three parameters:

- p: The autoregressive part is a linear regression that relates past values of data series to future values.
- d: The integrated part indicates how many times the data series has to be differentiated to get a stationary series.
- q: The moving average part that relates past forecast errors to future values of data series.

The practical and pragmatic approach of Box–Jenkins methodology in order to build ARIMA models is based on the following steps: (1) Identification, (2) Parameter Estimation and Selection, (3) Diagnostic checking, (4) Model's use. Mathematically, the ARIMA (p,d,q) model would be expressed as Eq. (3):

$$W_t = \mu + \frac{\theta(B)}{\phi(B)} a_t \tag{3}$$

where:

- t: Time index;
- W_t: the difference of the variable of interest z_t;
- μ: Reference point of the process level;
- $\theta(B)$: Moving averages operator $\Theta(B) = (1 - \Theta_1(B_1) - \Theta_2(B_2) - \cdots - \Theta_q(Bq))$;

- $\phi(B)$: Auto regressive operator $\Phi(B) = (1 - \Phi 1(B1) - \Phi_2(B_2) - \cdots - \Phi_p(B_p))$;
- B^p: Reverse operator $B^p z_t - z_{t-p}$;
- a_t: Random error.

ARIMA (p,d,q) stands for Auto Regressive Integrated Moving Average and would be expanded as in Eq. (4)

$$W_t = \Theta_0 + \Phi_1 W_{t-1} + \cdots + \Phi_p W_{t-p} + a_t - \Theta_1 a_{t-1} - \cdots - \Theta_q a_{t-q} \quad (4)$$

Where:

$$\Theta_0 = \mu\left(1 - \Phi_t - \cdots - \Phi_p\right) \quad (5)$$

2.2. Random forest

Random Forest (RF) is a Machine Learning Algorithm based on Decision Trees. This algorithm lies in the class of ensemble classifiers, and it has gained a significant interest in the recent past, due to its quality performance in several areas [41–43]. Given a training dataset, $Z = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, the RF creates a set of B bootstrap samples, created from a random re-sampling on the training set itself [44–47]. For each bootstrap sample Z_b, $b = 1, 2, \ldots, B$, a decision tree is constructed. When building these decision trees, each time a split is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors. When growing the tree, the idea behind selecting a random sample of $m < p$ predictors to consider in each step leads to uncorrelated trees from each sample. In this way, the bias increases in exchange for a decrease in variance. Let $g^b(x)$ be the prediction function obtained from the bth tree. The Random Forest technique a strong method that if its design be in a correct way, it would get a suitable result [43,48,49]. The prediction function obtained by the Random Forest algorithm is expressed by Eq. (6).

$$g_{rf}(x) = \frac{1}{B}\sum_{b=1}^{B} g^b(x) \quad (6)$$

2.3. Support vector regression

Support Vector Regression (SVR) is a powerful method for building a classifier for classification and regression first identified by Vladimir Vapnik and his research team in 1992 [50]. The SVR algorithm creates a decision boundary, known as the hyperplane, between two classes that allow predicting labels from one or more feature vectors. This decision boundary is oriented such that it is as far as possible from the closest data points from each of the classes. These closest points are called support vectors. Given a labeled training dataset, $Z \equiv (x_1, y_1), (x_2, y_2), \ldots, (x_B, y_n)$, $x_1 \in R^d$ and $y_i \in (-1, +1)$. The optimal hyperplane can then be defined as Eq. (7):

$$wx^T + b = 0 \quad (7)$$

where w is the weight vector, x is the input feature vector, and b is the bias. The hyperplanes would be described by Eqs. (8) and (9) respectively:

$$wx_i^T + b \geq +1 \text{ if } y_i = 1 \quad (8)$$

$$wx_i^T + b \leq -1 \text{ if } y_i = -1 \quad (9)$$

The objective of training an SVR algorithm is to find w and b in order that the hyperplanes separates the data and maximizes the margin $1/\|w\|^2$. Originally proposed to construct a linear classifier, the SVR algorithm would be used to model higher dimensional or nonlinear models using the kernel function [49,51]. For instance, in a non-linear problem, the kernel function would be used to provide additional dimensions to the raw data and turn it into a linear problem in the resulting higher dimensional space. The kernel function would provide faster calculations which need computations in high dimensional space. The kernel is an internal product, which would be linear or nonlinear(Polynomials), Radial-Basis Function(RBF), and the Sigmoid. It is defined as Eq. (10):

$$K(x, y) = \langle f(x), f(y)\rangle \quad (10)$$

2.4. WaveNet

WaveNet is a neural network based on the Convolution Neural Network (CNN) architecture. To better understand the WaveNet, the concept of origin and how CNN works is necessary. The CNN are algorithms that would be considered as bioinspired [52]. By Using CNN model, Fukushima (1980) proposes the neocognitron which follows the concept of hierarchical structure [53]: lateral geniculate body(LGB) $\longrightarrow$ simple cells $\longrightarrow$ complex cells $\longrightarrow$ hypercomposite cells lower order $\longrightarrow$ higher order hypercomplex cells. Thus, larger receptive fields are more insensitive to the change of stimulus pattern, being the cells of greater order and greater complexity that responds more selectively. Neocognitron is able to recognize patterns in the image very similar to convolutional neural networks, however, if the concept of convolution between the layers is used. Although CNN is famous for image classification and object recognition problems, recently this kind of neural networks has been employed for time series analysis. The model is almost the same, but some modification may be necessary (one dimensional convolution neural network). In the literature, other works have already explored CNN as a forecaster for time series stock prices, such as trend prediction with CNN of two dimensions [54]. WaveNet is a variation of the CNN architecture created to solve text-to-speech problems [55]. That architecture does not rely on neurons and activation functions but only on weights of the filters of each convolution layer. The window of weights is slide across the input series. The WaveNet relies on a causal structure, as show in Fig. 1.

2.5. LSTM

LSTM networks are an adjustment of *RNN* in special the conversion is concentrated on the hidden layer of the network [37]. Fig. 2 shows the general structure of LSTM, including input cell, hidden layers, and output cell.

The first step in the LSTM structure is to be selected what information is thrown away from the cell state. This decision is performed by a sigmoid layer termed the "forget gate layer". In detail, h_{t-1}, x_t, and outputs represent a number between 0 and 1 for each number in the cell state c_{t-1}. This step denotes "completely keep this" while the previous step signifies "completely get rid of this", as it is shown in Eq. (11).

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t] + b_r\right). \quad (11)$$

The next step is to choose what new information will be saved in the cell state, which has two parts. First, there is a sigmoid layer called the "input gate layer" determines which values will update. In the following, a *tanh* layer builds a vector of new candidate values, $\tilde{c}_t$, that would be joined the state which are represented in Eqs. (12) and (13).

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right). \quad (12)$$

$$\tilde{c}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right). \quad (13)$$

Subsequently, the old cell state will update, c_{t-1}, into the new cell state c_t. After that, the old state is multiplied by f_t, skipping the

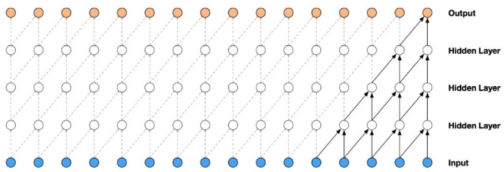

Fig. 1. Architecture proposed by the WaveNet authors [55].

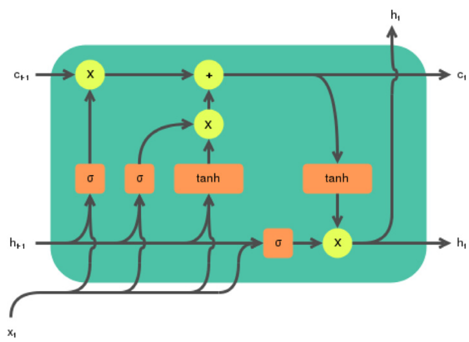

Fig. 2. LSTM cell state.

things which were decided to be forgotten earlier. Then $i_t \times \tilde{c}_t$ is added. These are the new candidate values, estimated by how much those w are decided to update each state value which is represented in Eq. (14).

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t. \tag{14}$$

Finally, the prospective outcome shall be decided. This output will be based on the cell state but will be filtered. First, the sigmoid layer is run which determines what parts of the cell state are going to be the output. Then, the cell state is put through tanh (to push the values to be between -1 and 1) and multiply it by the output of the sigmoid gate. Thus, only these parts of the output have been selected, which are represented in Eqs. (15) and (16).

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right). \tag{15}$$

$$h_t = o_t \times \tanh\left(c_t\right). \tag{16}$$

Old data goes through the entire network, improving its states to obtain the expected prediction. In the LSTM method, the effects of sudden events in the sequence will be high. For example, on seven consecutive days, if negative news is received that is denied immediately, there would exist a sudden fall in the price of Bitcoin, which rises again instantly. In fact, there is a noise, but this noise in the next day's prediction with The LSTM method will be very effective. To reduce this error a supervised MLP structure would be used in such a way that the effect of such noises is less considered.

2.6. Other methods

In general, variety of methods have been used to predict the price of cryptocurrencies particularly Bitcoin, some of which are presented in Table 1.

3. Proposed method: Bitcoin price prediction based on learnable window size(LWS)

In this paper, MLP deep neural network structures are used in a consecutive and intelligent manner. As it is mentioned earlier, the price of Bitcoin depends on not only the features affecting its price the day before but also an indefinite period of time from the previous days' data would have a significant impact. With respect to the approach adopted, a window containing the data of the previous days affecting the price of Bitcoin is provided intelligently and according to the fluctuations. Thus, the price of would be predicted more accurately for the next day. Fig. 3, is showing the proposed system model.

Firstly, the most effective features in Bitcoin price prediction are recognized. In the second part, data is created based on two different sections including daily value and the difference in the level of values between the previous day and the day before the previous day. Then data is normalized and prepared to go for the prediction part. In the next stage, Learnable Window Size-MLP (LWS-MLP) as a primary deep neural network will be used to predict the best window size based on the conditions in the previous days. It means seven sets of prediction data for one-day to seven-day window are created, and then each set that has better efficiency in Bitcoin price prediction as a best window size is applied on the secondary deep neural network structure. Fig. 4, is showing LWS-MLP structure which is used in this paper.

3.1. Learnable window size model

As it is mentioned briefly, a primary deep neural network is trained based on bitcoin price fluctuations to select the best window size. This primary neural network can include seven different modes of one-day to seven-day windows. Therefore seven MLP_i are recorded, and their related dimensions are from 4 to 16 including Bitcoin price and the Bitcoin price difference between previous day and the day before previous day, respectively. After predicting each of the seven MLP_i, the best result is chosen as the ideal window in the primary MLP which is formulated in Eq. (17).

$$LWS_t = \text{Min} \ \left| \ \text{P}_{\ t+1} - MLP_{i,t} \ \right|. \tag{17}$$

Where P_{t+1} is the real Bitcoin price in day $t+1$, $MLP_{i,t}$ is the predicted result of $i - day$ window MLP for day t+1, and LWS_t is the best window size to be used in the secondary MLP neural network.

In Fig. 5, numbers from one to seven are respectively one- to seven-day window sizes including 42 effective features.

3.2. Learnable window size prediction

In this method, the 42 input data for each day, which are generated as vectors for seven consecutive days, predict an output as the best price for the next day after the LWS neural network predicted the best window size. Algorithm 1 illustrates this procedure.

In the process of Algorithm 1, w, b, and i are respectively weights, bias, and a number of Hidden Layers of the neural MLP network to show the Learnable Window Size process. Also, w', b', and i' are respectively weights, Bias, and a number of Hidden Layers neural MLP networks, to perform the final prediction of Bitcoin. The number of hidden layers is equal for all types of windows. Also, m the number of each of the seven MLPs is based on seven different window sizes, and if LWS is predicted for each of the m, their system will go to the main output to predict the price of Bitcoin. The proposed method is highly robust during the system process against potential noises because sudden changes do not have strong support to create a significant effect on the price of Bitcoin. Fig. 6 shows an overview of the proposed system.

Table 1
Some research on cryptocurrencies.

Author	Year	Explain	Technique	Outcomes	Cryptocurrencies	Resource
[56]	2020	They provided a prediction model with the ability to estimate Bitcoin price	Deep learning, deep recurrent convolutional neural networks(DRCNN)	The percentage of accuracy is higher in the DRCNN (94.19%) rather than DL	Bitcoin	IMF's International Financial Statistics (IFS), the World Bank, FRED Sant Louis, Google Trends, Quandl, and Blockchain.info
[57]	2020	The effect of geopolitical risk, global and US economic policy uncertainty on the structure of Bitcoin correlation with various Financial and commodities asset classes was perused.	GJR-GARCH models	Geopolitical risk is the most important feature that donates to the volatility and risk premia of Bitcoin.	Bitcoin	Coindesk.com
[58]	2020	They classified Bitcoin price by daily and high-frequency price.	Machine learning algorithms	The LR model gained an accuracy of 66% in forecasting Bitcoin price.	Bitcoin	CoinMarketCap.com
[59]	2020	They built a feature system with 40 determinants that affect the price of Bitcoin.	Stacked Denoising Auto Encoders (SDAE), Back Propagation Neural Network (BPNN), Support Vector Regression (SVR)	SDAE achieved the lowest MAPE and RMSE of 0.1019 and 160.63	Bitcoin	Www.coindesk.com and BTC.com.
[33]	2021	They addressed predict of Bitcoin price	LSTM and Gated Recurrent Unit (GRU)	In LSTM, RMSE is 0.092 and in GRU, RMSE is 0.075.	Bitcoin, Ripple, Ethereum, Ethereum classic, Lite coin	https://www.kaggle.com
[60]	2021	They suggested a differential evolution-based regression structure for predicting one day ahead price of Bitcoin	SVR, Polynomial Regression with Interaction (PRI), MLP	Nash–Sutcliffe efficiency (NSE) is 0.998593	Bitcoin	Blockchain info
[61]	2021	They considered thirteen attributes for Bitcoin price forecasting and carried nonlinear Autoregressive with external input analysis	Levenberg–Marquard, Bayesian Regularization, and Scaled Conjugate Gradient algorithm	The Bayesian Regularized Neural Network presented the better result than previous.	Bitcoin	https://santiment.net/
[62]	2021	They examined the global Bitcoin price shifting trends due to social media communication data	Latent Dirichlet Distribution (LDA) model	Social media had positive effect on Bitcoin price	Bitcoin	https://bitcointalk.org/discussion
The proposed method	2021	Provide a new learnable Window and structure for Bitcoin price prediction	MLP (learnable Window Size)	MAPE error 0.310048417for mid of 2017–2019 and 6.269789757 for 2019–2021 (May)	Bitcoin	www.investing.com, www.blockchain.com, trends.google.com/trends

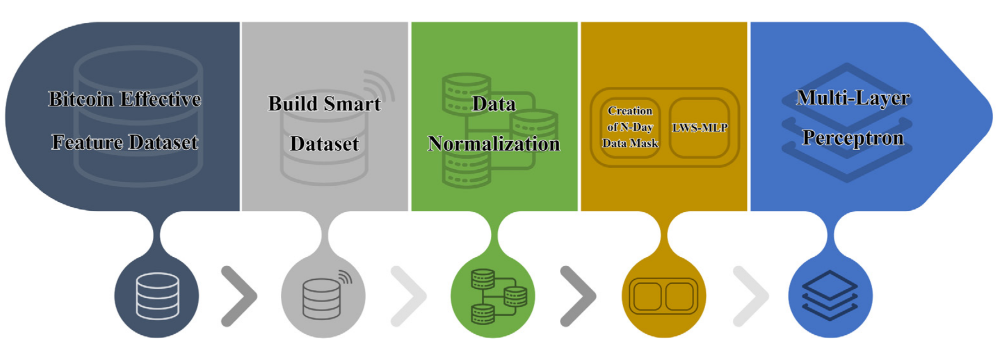

Fig. 3. The proposed system model.

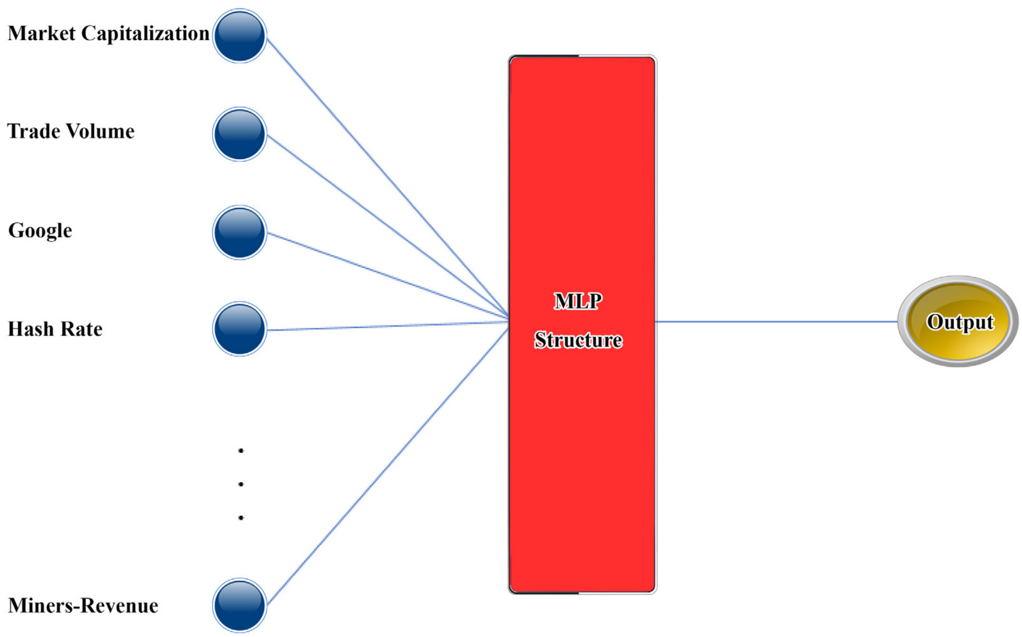

Fig. 4. MLP structure to predict Bitcoin price.

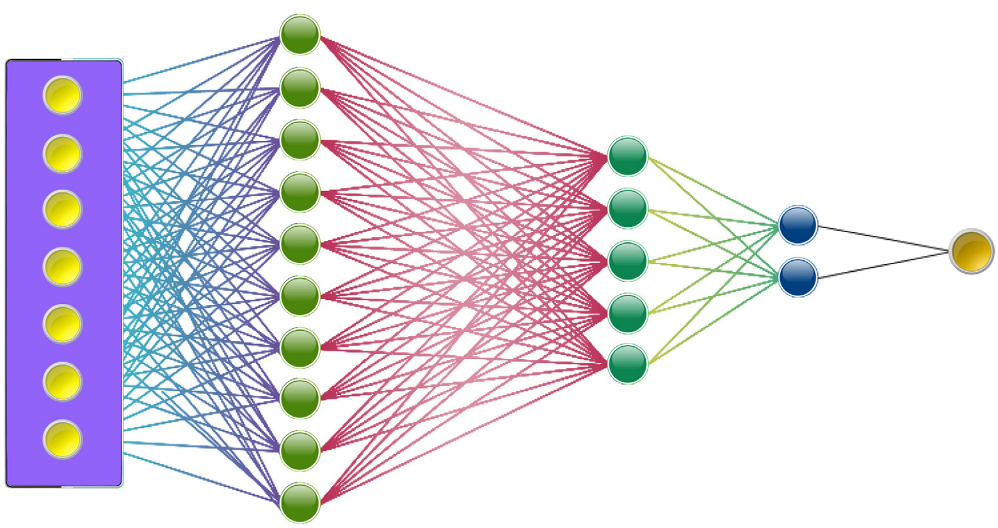

Fig. 5. Learnable window size multi-layer perceptron structure (LWS-MLP).

4. Results and discussion

In this section, a description of the features contributing to predicting the price of Bitcoin is presented. Moreover, the pre-processing task of the data is explained. Then, the evaluation metrics used to measure the performance of the proposed model is introduced and the results will be compared to the other methods.

4.1. Dataset

Many studies have been done to show the correlation between the variables affecting the price of Bitcoin. These include examining the relationship between the Bitcoin market price and market value, market sentiment such as volume of tweets and Google, Bitcoin trading volume, and Bitcoin mining [18,19]. In addition, the relationship between network stiffness, network security, the number of miners, blocks and, Bitcoin price have been determined [12]. In this study, there are 21 decision variables to predict the price of Bitcoin. A total of 1275 training dataset used in the proposed model to extract patterns of Bitcoin price. On the other hand, for each decision variable, a new feature is created that includes the difference between previous day compared to the day before previous day which is obtained through Eq. (18). This allows the 21 decision variables to convert to a dataset with 42 features per day.

$$f_i^j = f_{i-1}^j - f_{i-2}^j. \tag{18}$$

Where f is the intended feature, j is the number of intended features and i is the number of intended days. Therefore, the feature vector of X in this paper to train and test deep learning system, is created as algorithm 2.

Eq. (19) is a linear regression model to calculate the correlation between these 42 features and Bitcoin price daily trend.

$$\mathrm{Corr}_f = \sum_i \frac{|\ \mathrm{P}_{i+1} - \mathrm{P}_i\ |}{\mathrm{P}_{i+1} - \mathrm{P}_i} \times \frac{|f_{i+1} - f_i|}{f_{i+1} - f_i}. \tag{19}$$

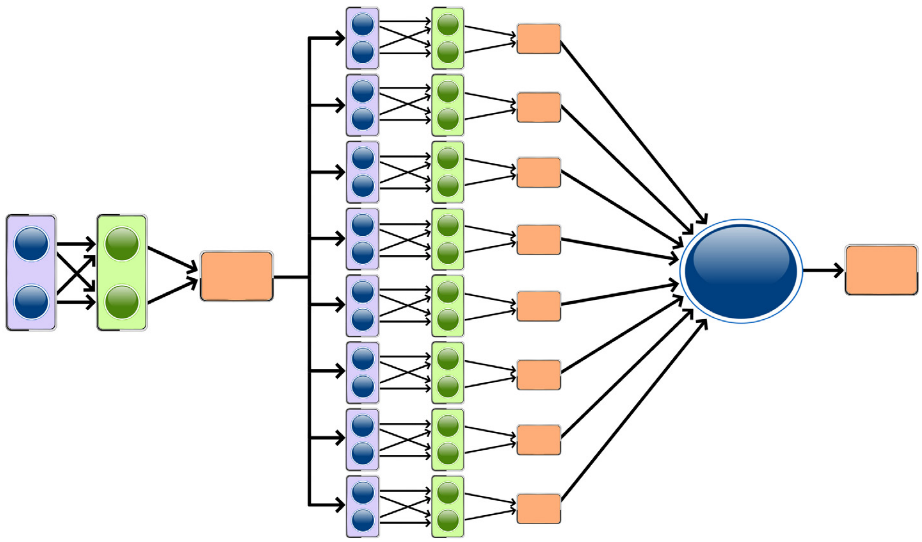

Fig. 6. Architecture of prediction based on learnable window size.

Algorithm 1 Bitcoin price Prediction

procedure PREDICTION
Input : X, *Market indicators and Bitcoin data.*
Input : *Model architecture having* $l_L WS$ *hidden layers.*
Input : *Model architecture having* l_P *hidden layers.*
Input : *Trained* $\boldsymbol{w}^i, \boldsymbol{b}^i \quad \boldsymbol{i} \in \{\mathbf{1}, \dots, \boldsymbol{l}_{LWS}\}$
$\boldsymbol{w}'^{i'}_m, \boldsymbol{b}'^{i'}_m \quad \boldsymbol{i}' \in \{\mathbf{1}, \dots, \boldsymbol{l}_P\}, \boldsymbol{m} \in \{\mathbf{1}, \dots, 7\}$
Output : $LWS_t, P_t \quad t \in \{1, \dots, n\}$
Forward Propagation :
for $t = 1$ *to* n *do*
$s^0_{t,7} = X_t^{\max(m)}$
for $k = 1$ *to* LWS_t *do*
$z^k_{t,\max(m)} = b^k_{t,\max(m)} + W^k_{t,\max(m)} s^{k-1}_{t,\max(m)}$
$\boldsymbol{h}^k_{t,\max(m)} = \text{activation}\left(z^k_{t,\max(m)}\right)$
end for
$LWS_t \leftarrow h_t^{LWs}$
$s^0_{t,LWS_t} = X_t^{LWS_t}$
for $k' = 1$ *to* l_p *do*
$z'^{k'}_{t,LWS_t} = b'^{k'}_{t,LWS_t} + W'^{k'}_{t,LWS_t} s^{k'-1}_{t,LWS_t}$
$\boldsymbol{h}'^{k'}_{t,LWS_t} = \text{activation}\left(z'^{k'}_{t,LWS_t}\right)$
end for
$\boldsymbol{P_t} \leftarrow \boldsymbol{h}'^{l_p}_{\boldsymbol{t},LWS_t}$

Algorithm 2 Vector neural network training and testing feature

procedure BUILDING DATASET
$Counter = 1$
for t *in range* $(Dataset)$
for i *in range* $(Feature * 2)$
$X_t^{\text{Counter}} = F_q^i$
$X_t^{\text{Counter }+1} = F^i_{q-1} - F^i_{q-2}$
$Counter = Counter + 2$
end for
end for

Where the P is Bitcoin price, and f_i is the value of ith feature. The detailed results are shown in Table 2.

This paper tries to access the most influential variables based on Linear correlation analysis. There are totally 26 feature variables which just a few of them have been eliminated. The correlation rate of more than 0.3 is a notable feature in Bitcoin price prediction which shows that we consider easy structure to select feature variables. Besides, by considering Prediction Hardship Factor (PHF), it is demonstrated that our model is robust in contrast of harsh fluctuations.

4.1.1. Dataset normalization

In general, the dataset should be normalized because the features intended for Bitcoin price prediction do not have the same unit and range. The method which is used for normalization in this study is mean-normalization. Eq. (20) shows the mean-normalization formula.

$$\|f_i\| = \frac{f_i - \mu_i}{\max(f) - \min(f)}. \tag{20}$$

Where f_i is the ith feature, μ_i is the average of ith feature in the corresponding period and $\|f_i\|$ is a normalized feature of f_i.

4.2. Evaluation metrics

According to evaluation model, the error is evaluated through mean absolute percentage error (MAPE), Root Mean Squared Error (RMSE) which are formulated as Eq. (21):

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{|y_t|} \times 100 \tag{21}$$

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

where y_t is the actual values and $\hat{y}_t$ are the forecasted values. Also, Prediction Hardship Factor(PHF) as a market price fluctuations rate has been introduced to measure the difficulty and volatility of datasets which is formulated in Eq. (22).

$$\text{PHF} = \sqrt{\frac{1}{n}\left|\sum_{t=1}^{n}(y_t - y_{t-1})^2\right|} \tag{22}$$

Where y_t is original price in day t, $\hat{y}_t$ is predicted price related to day t, and n is the number of total backtesting data.

Table 2
Correlation of variables with Bitcoin price.

Variable	Correlation	Define variables
Market capitalization	0.99412	It shows the total value of the Bitcoin market in US dollars.
Trade-volume	0.39532	The total volume of Bitcoin transactions made.
Transaction-fees-USD	0.57825	It shows the total cost of extracting a Bitcoin in dollars.
Average-confirmation-time	0.39837	The average time it takes for a transaction to be accepted in the extracted block and placed in the public ledger.
Difficulty	0.62627	It is a variable whose purpose is to keep the average time of creating a block in the network constant.
High price	0.99848	The highest price of Bitcoin is per day
Low price	0.99751	The lowest price of Bitcoin per day
Total hash-rate	0.6152	It is a measure of the performance of a minor device. In other words, the hash rate indicates the rate at which a miner succeeds in solving the hash to receive revenue.
Block-size	0.62398	The total size of transactions accepted in each block
Miners-revenue	0.85112	The revenue received by the miners is for building each block.
N-transactions-total	0.61815	The total number of transactions in the blockchain
Google	0.34686	The number of Bitcoin searches by people on Google
Open price	0.99962	The initial price of Bitcoin is per day.
N-payments-per-block	0.48574	It is the total number of payments per block over the past 24 h.
Total circulating Bitcoin	0.31442	The total number of mined Bitcoin that are currently circulating on the network
Cost-per-transaction-percent	−0.34551	Percentage of miners' revenue they receive from their trading volume.
Fees-USD-per-transaction	0.6111	It shows the average cost of Bitcoin mining per trade in dollars.
N-unique-addresses	0.5893	It is the total number of transactions, excluding those involving the network's 100 most popular addresses.
N-transactions-per-block	−0.37992	The total number of transactions made in each block
Output-volume	0.3557	It is the total value of all transaction outputs per day. This includes coins returned to the sender as change.

4.3. Statistical results

In this section, the results obtained by the deterministic models are illustrated. The architecture of the model for the primary MLP structure which is defined to predict window size is as following.

The trained MLP model contains 4 layers, each with the same activation function ReLU, and trained using the Adam algorithm for 300 epochs. The input consists of price and the daily variation of Bitcoin price over the past 7 days. The layers contain 10, 5, 2, 1 neurons hierarchically in the input to the output direction.

For the secondary MLP neural network, the trained MLP model contains 7 layers, each with the same activation function ReLU, and trained using the Adam algorithm for 800 epochs. The input consists of 42 features over the specific time period based on the window size predicted from the previous step. The layers contain 200, 150, 90, 50, 15, 10, 1 neurons hierarchically in the input to the output direction. The proposed method is implemented on a system with the following specifications (CPU:2.3 GHz core i5, RAM:4 GB) which shows does not need to high professional configuration. Each epoch of primary and secondary neural network learning took 0.05 and 0.2 s respectively. Also, this system has been simulated on Jupyter notebook which is a powerful compiler under Python language. In the following, the prediction data are compared by considering the two, four, five, six, seven-day windows and LWS-MLP for prediction the next day. Fig. 7 shows a comparison of fixed-window sizes-MLP, LWS-MLP and original price since 12 Nov 2018 to 30 Apr 2019 which is included 170 backtesting data. It is worth to mention that the backtesting data is not used on training procedure.

Also, to illustrate the most effective window size, Fig. 8 shows the number of times which a fixed window size was the best one in terms of Bitcoin price prediction.

As it is clear in Fig. 8, considering 5 days in advance to predict tth day with 46% accuracy had the greatest impact on the optimal window size for prediction. To compare and evaluation of our system, various of recent methods which have been implemented on this area considered such as ARIMA, RF, SVR, LSTM stochastic and MLP stochastic [19], Wavenet. For selecting the parameters for the ARIMA model, the Auto ARIMA procedure was used. The Auto ARIMA approach is used to perform a grid search over multiple values of p, d, q considering the AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) criteria. These generated values are used to determine the best combination of parameters. For the Random Forest model, we chose the hyperparameters to provide a better performance in the segmentation of the data so as not to obtain an over fitted model. The hyperparameters varied were: n estimators, which representing the number of trees in the forest; criterion, which corresponds to the function to measure the quality of a split; max depth, the maximum depth of the tree; min samples split, the minimum number of samples required to split an internal node; min samples leaf, the minimum number of samples required to be at a leaf node; and max features, the number of features to consider

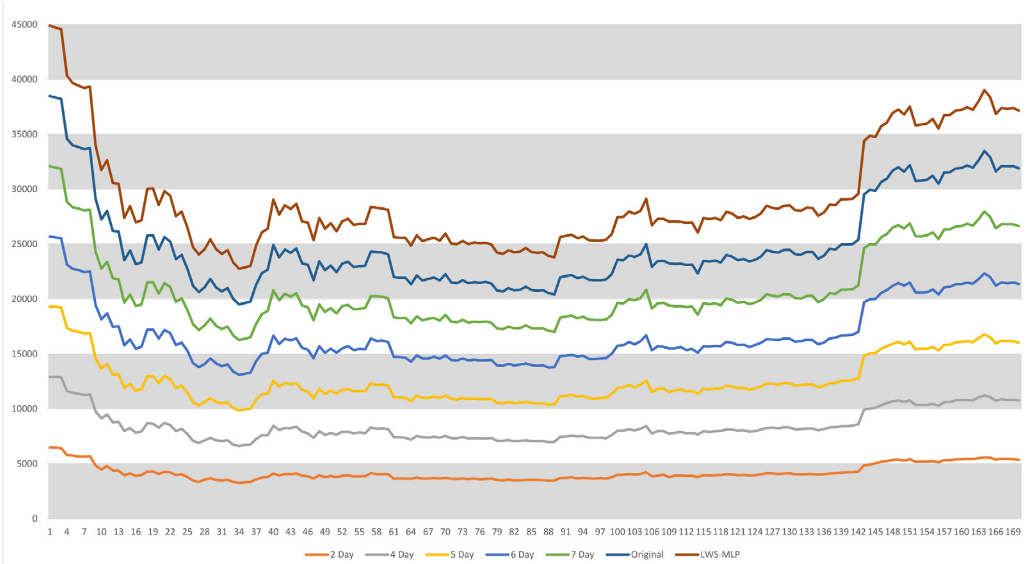

Fig. 7. Comparison between Original prices, fixed and learnable window size predictions(12 Nov 2018–30 Apr 2019).

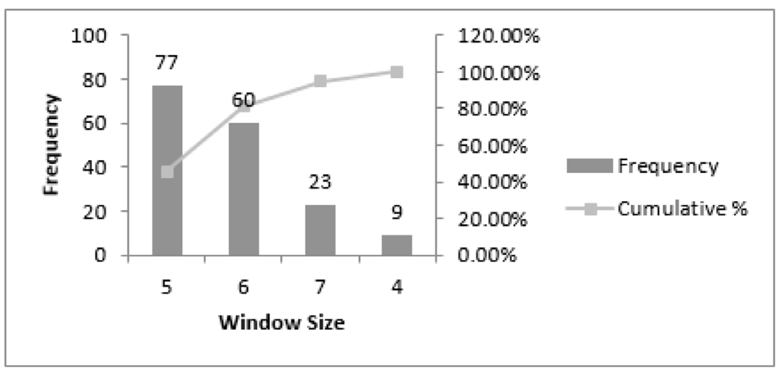

Fig. 8. Histogram of the number of times which a fixed window size was the best one for prediction(12 Nov 2018–30 Apr 2019).

when looking for the best split. In the Support Vector Machine model, our choices of variable hyper-parameters were: kernel, which specifies the kernel method to be used in the algorithm for dividing the data edges; degree, degree of the polynomial kernel, but these are ignored by all the other kernels; gamma and C, responsible for defining a large or small edge between data segmentation. The input and output of the WaveNet model is the same of LSTM. We used a one-dimensional WaveNet for the time series prediction. All the hyper-parameters used for LSTM are also used for CNN. There are two main differences: the filter size and the number of convolutions. In each layer we made fewer convolutions since the main characteristics are extracted from the latest layers Regarding the padding, we adopted a padding that results in an output of the same size as the input. The activation function adopted is the Leaky ReLU. We added a batch regularization to improve the generalization power of the model. The convolutions occur with the one-dimensional vector with the size of the filter. Table 3 shows the comparison between mentioned and proposed methods since 12 Nov 2018 to 30 Apr 2019 including 170 backtesting data.

By paying attention to Table 3, MAPE and RMSE as proper measurements to compare various methods is significantly lower in LWS-MLP compared to other methods under similar situation. To show the results in a clear way, the comparison line chart is demonstrated in Figure 9.

In an another analysis, the MAPE and RMSE error metric evaluations are calculated under harsh situation (16 Dec 2020 to 4 Jun 2021) with high PHF almost 1004, however, PHF in previous duration (12 Nov 2018 to 30 Apr 2019) was just almost 387.

By paying attention to Table 4, MAPE and RMSE metrics again in LWS-MLP method compared to other methods has noticeably better performance under same situation. This shows the excitement and fluctuations of the market is significantly higher in this duration. In general, the pattern of behavior created based on the method presented in this paper is strong even in severe market fluctuations and creates a small amount of MAPE and RMSE. Furthermore, the comparison line chart is illustrated in Figure 10. By considering the fluctuation behavior of the Bitcoin price market, the PHF is presented in Eq. (22) MAPE and RMSE value is completely acceptable which would be used to evaluate Bitcoin market.

Table 3
MAPE in Bitcoin price prediction in LWS-MLP Vs. other methods since 12 Nov 2018 to 30 Apr 2019.

Model	MAPE	RMSE
ARIMA	6.43	684.385
RF	11.37	1099.770
SVR	2.92	282.115
LSTM Stochastic	3.18	327.782
MLP Stochastic	2.55	246.253
WaveNet	14.31	1475.56
LWS-MLP	**0.31**	**48.918**

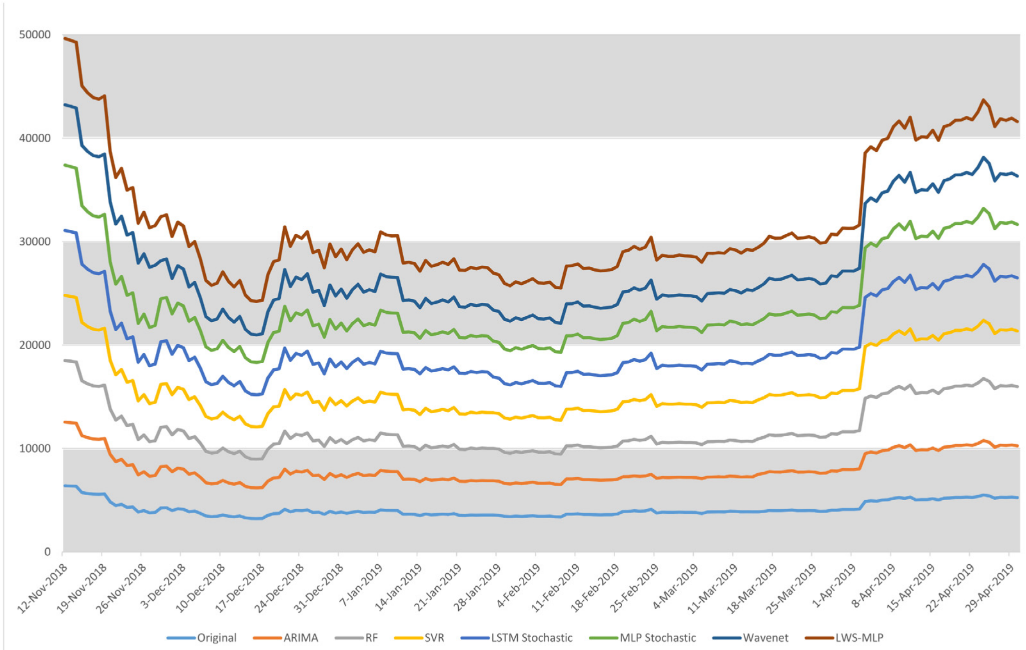

Fig. 9. Trendline in Bitcoin price Prediction in LWS-MLP Vs. other methods since 12 Nov 2018 to 30 Apr 2019.

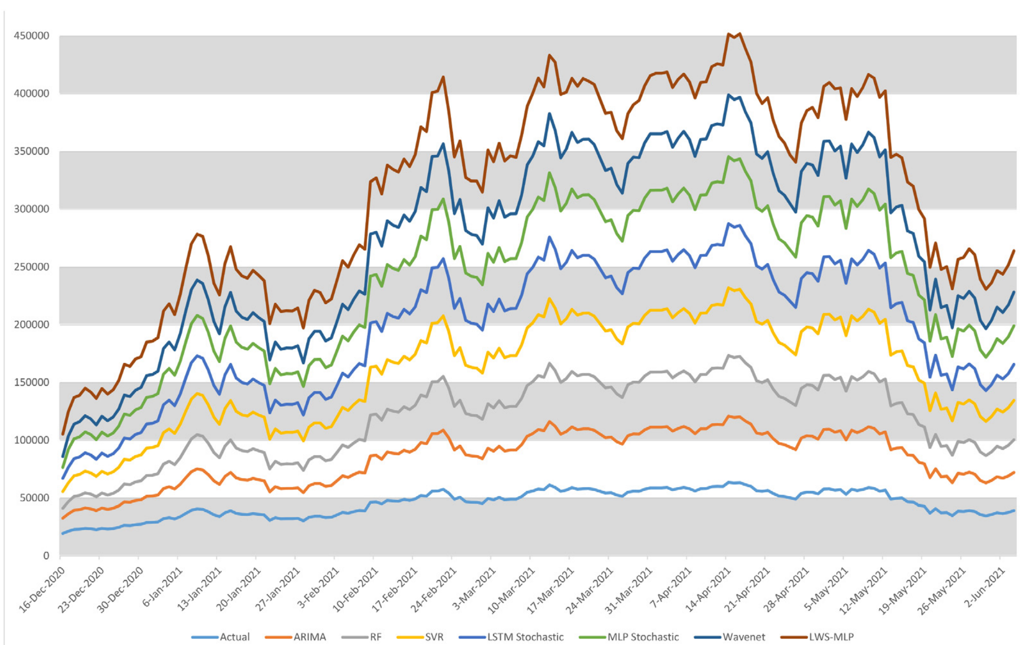

Fig. 10. Trendline in Bitcoin price Prediction in LWS-MLP Vs. other methods since 16 Dec 2020 to 4 Jun 2021.

Table 4
MAPE in Bitcoin price prediction in LWS-MLP Vs. other methods since 16 Dec 2020 to 4 Jun 2021.

Model	MAPE	RMSE
ARIMA	14.8533	14858.856
RF	26.7195	26298.864
SVR	11.972	11954.030
LSTM Stochastic	19.3176	19126.447
MLP Stochastic	13.056	13642.156
WaveNet	23.2815	23908.058
LWS-MLP	**6.2697**	11323.790

5. Conclusion

With the passage of time and the acceptance of Bitcoin in societies more than before, the tendency to investigate this currency is growing. On the other hand, the risk of investigation in the Bitcoin market is the cause of precautions in investors' actions. It is required that methods be selected to reduce potential risks which can have high accuracy in price prediction. Therefore, this paper presents a new method for dealing with this risk and to perform an almost exact prediction of Bitcoin price, even in high fluctuation situations of the market. In this paper, influential variables on Bitcoin price are presented, and by considering different variables and their correlation with Bitcoin price trend, the most critical features are selected to train the deep neural network. In the following, a primary MLP neural network structure was provided in order to predict the best window size for Bitcoin price prediction and to detect the influence of noise-type behaviors. The Prediction Hardship Factor (PHF) was also introduced in this paper to show under what conditions and with what fluctuation the calculated MAPE was obtained. The evaluations have shown that this method has been able to get the Mean Absolute Percentage Error (MAPE) of about 6.26% under harsh situations(16 Dec 2020 to 4 Jun 2021) which shows an absolutely acceptable amount. In future research, this method would be used to predict other cryptocurrencies such as Ethereum, Litecoin, gold coin, etc.

CRediT authorship contribution statement

Shahab Rajabi: Conceptualization, Methodology, Simulation, Visualization, Review & editing. **Pardis Roozkhosh:** Data analysis and collection, Simulation, Investigation, Writing – review, Validation. **Nasser Motahari Farimani:** Supervision, Investigation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Nasser Motahari Farimani reports financial support was provided by Ferdowsi University of Mashhad. Nasser Motahari Farimani reports a relationship with Ferdowsi University of Mashhad that includes: funding grants.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported in part by: Research Deputy of Ferdowsi University of Mashhad, under Grant No. 55646 (dated September 12, 2021).

References

[1] F. Steinmetz, M. Meduna, L. Ante, Ownership uses and perceptions of cryptocurrency: Results from a population survey, Technol. Forecast. Soc. Change 173 (2021) http://dx.doi.org/10.1016/j.techfore.2021.121073.
[2] L. Malherbe, Cryptocurrencies and blockchain: Opportunities and limits of a new monetary regime, Int. J. Political Econ. 48 (2019) 127–152.
[3] T. Huab, X. Liub, T. Chena, Z. Xiaoming, H. Weina, Y. Liu, Transaction-based classification and detection approach for Ethereum smart contract, Inf. Process. Manage. 58 (2) (2021).
[4] A. Sai, J. Buckley, B. Fitzgerald, A. LeGear, Taxonomy of centralization in public blockchain systems: A systematic literature review, Inf. Process. Manage. 58 (4) (2021).
[5] G. Selgin, Synthetic commodity money, J. Financial Stab. (2015) 92–99.
[6] M. Mamun, Gazi Salah Uddin, Muhammad Tahir Suleman, Sang Hoon Kang, Geopolitical risk, uncertainty and Bitcoin investment, Physica A (2020) 609–735.
[7] Z. Zhang, H. Daid, J. Zhoua, M. Subrot Kumar, M. García, H. Wang, Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels, Expert Syst. Appl. 183 (2021) http://dx.doi.org/10.1016/j.eswa.2021.115378.
[8] J. Hendrickson, T. Hogan, W. Luther, The political economy of Bitcoin, Econ. Inq. 54 (2) (2016) 925–939.
[9] A. Dyhrberg, Bitcoin, gold and the dollar–A GARCH volatility analysis, Finance Res. Lett. (2016) 85–92.
[10] M. Nakano, A. Takahashi, S. Takahashi, Bitcoin technical trading with artificial neural network, Physica A 510 (2018) 587–609.
[11] W. Kristjanpoller, M. Minutolo, A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis, Expert. Syst. Appl. 109 (2018) 1–11.
[12] S. Das, M. Rou, A hybridized ELM using self-adaptive multi-population-based jaya algorithm for currency exchange prediction: an empirical assessment, Neural Comput. Appl. (2018) 1–24, http://dx.doi.org/10.1007/s00521-018-3552-8.
[13] Y. Shynkevich, T. McGinnity, S. Coleman, A. Belatreche, Y. Li, Forecasting price movements using technical indicators: Investigating the impact of varying in- put window length, Neurocomputing 264 (2017) 71–88.
[14] S. Lahmiri, Minute-ahead stock price forecasting based on singular spectrum analysis and support vector regression, Appl. Math. Comput. 320 (2018) 444–451.
[15] L. Lei, Wavelet neural network prediction method of stock price trend based on rough set attribute reduction, Appl. Soft Comput. 62 (2018) 923–932.
[16] Y. Kim, J. Lee, N. Park, J. Choo, J. Kim, C. Kim, When Bitcoin encounters information in an online forum: Using text mining to analyse user opinions and predict value fluctuation, PLoS One 12 (2017).
[17] B. Li, K. Chan, C. Ou, S. Ruifeng, Discovering public sentiment in social media for predicting stock movement of publicly listed companies, Inf. Syst. 69 (2017) 81–92.
[18] S. Lahmiri, S. Bekiros, Cryptocurrency forecasting with deep learning chaotic neural networks, Chaos Solitons Fractals 118 (2019) 35–40.
[19] P. Jay, V. Kalaria, P. Parmar, S. Tanvar, N. Kumar, M. Alaza, Stochastic neural networks for cryptocurrency price prediction, IEEE Access 8 (2020).
[20] A. Tiwari, A. Jana, Debojyoti Das, David Roubaud, Econom. Lett. (2018) 106–109.
[21] R. Jana, I. Ghosh, D. Das, A differential evolution-based regression framework for forecasting Bitcoin price, Ann. Oper. Res. 306 (2021) 295–320.
[22] K. Dhinakaran, P.Baby. Shamini, D. Indhumathi, R. Asha, Cryptocurrency exchange rate prediction using ARIMA model on real time data, in: International Conference on Electronics and Renewable Systems, ICEARS, 2022.
[23] G. Nikol, D. Kukolj, R. Adcock, V. Djakovic, Forecasting Bitcoin with technical analysis: A not-so-random forest? Int. J. Forecast. (2021).
[24] H. Guo, D. Zhang, S. Liu, L. Wang, Y. Ding, Bitcoin price forecasting: A perspective of underlying blockchain transactions, Decis. Support Syst. (2021) http://dx.doi.org/10.1016/j.dss.2021.113650.
[25] Y. Ma, Z. Luan. Ethereum synchronicity, Upside volatility and Bitcoin crash risk, Finance Res. Lett. (2021) http://dx.doi.org/10.1016/j.frl.2021.102352.
[26] A. Khedr, I. Arif, M. El-Bannany, S. Alhashmi, M. Sreedharan, Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey, Intell. Syst. Account. Finance Manag. 28 (1) (2021) 3–34, http://dx.doi.org/10.1002/isaf.1488.
[27] A. Bariviera, The inefficiency of Bitcoin revisited: A dynamic approach, Econom. Lett. (2017) 1–4.
[28] J. Alvarez-Ramirez, E. Rodriguez, C. Ibarra-Valde, Long-range correlations and asymmetry in the Bitcoin market, Physica A (2018) 948–955.
[29] O. Berat, S. Mehmet, U. Gudelek, A. Murat, O. oglu, Financial time series forecasting with deep learning : A systematic literature review: 2005–2019, Appl. Soft Comput. 90 (2020) http://dx.doi.org/10.1016/j.asoc.2020.106181.
[30] M. Rosenstein, J. Collins, De. Luca, A practical method for calculating largest Lyapunov exponents from small data sets, Physica D 65 (1993) 117–134.
[31] E. Koo, G. Kim, Prediction of Bitcoin price based on manipulating distribution strategy, Appl. Soft Comput. 110 (2021) http://dx.doi.org/10.1016/j.asoc.2021.107738.
[32] W. Chen, H. Xu, L. Jia, Y. Gao, Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants, Int. J. Forecast. (2021).
[33] T. Awoke, M. Rout, L. Mohanty, S.C. Satapathy, Bitcoin price prediction and analysis using deep learning models, Commun. Softw. Netw. (2021) http://dx.doi.org/10.1007/978-981-15-5397-4-63.
[34] S. Cavalli, M. Amoretti, CNN-based multivariate data analysis for Bitcoin trend prediction, Appl. Soft Comput. 101 (2021) http://dx.doi.org/10.1016/j.asoc.2020.107065.
[35] K. Nguyen, The correlation between the stock market and Bitcoin during COVID-19 and other uncertainty periods, Finance Res. Lett. (2021) http://dx.doi.org/10.1016/j.frl.2021.102284.
[36] S. Choi, J. kShin, Bitcoin: An inflation hedge but not a safe haven, Finance Res. Lett. (2021) http://dx.doi.org/10.1016/j.frl.2021.102379.
[37] A. Bejaoui, N. Mgadmi, W. Moussa, T. Sadraoui, A short-and long-term analysis of the Nexus between Bitcoin, social media and COVID-19 outbreak, Heliyon 7 (7) (2021) http://dx.doi.org/10.1016/j.heliyon.2021.e07539.
[38] X. Guo, F. Lu, Y. Wei, Capture the contagion network of Bitcoin – Evidence from pre and mid COVID-19, Res. Int. Bus. Finance 58 (2021) http://dx.doi.org/10.1016/j.ribaf.2021.101484.
[39] V. Subashini, R. Janaki, G. Valarmathi, S. Suganthi, R. Prabha, K. Sivasankari, S. Kavitha, Survey on regression analysis of photoplethysmography using machine learning, Mater. Today Proc. (2021) 3743–3748, http://dx.doi.org/10.1016/j.matpr.2021.02.013.

[40] G. Box, G.M. Jenkins, Time series analysis: Forecasting and control, Holden-Day (1976).
[41] M.L. Calle, V. Urrea, Letter to the editor: Stability of random forest importance measures, Brief. Bioinform. 12 (1) (2011) http://bib.oxfordjournals.org/cgi/content/full/12/1/86.
[42] X. Chen, H. Ishwaran, Random forests for genomic data analysis, Genomics 99 (6) (2012) 323–329.
[43] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.
[44] J. Han, M. Kamber, J. Pei, Data Mining Concepts and Techniques, third ed., Waltham, Mass, 2012.
[45] M. Mudassir, S. Bennbaia, D. Unal, M. Hammoudeh, Time-series forecasting of Bitcoin prices using high-dimensional features: A machine learning approach, Neural Comput. Appl. (2020) http://dx.doi.org/10.1007/s00521-020-05129-6.
[46] K. Rathan, S. Sai, S.T. Manikanta, Crypto-currency price prediction using decision tree and regression techniques, in: 2019 3rd International Conference on Trends in Electronics and Informatics, ICOEI, 2019.
[47] A. Ibrahim, R. Kashef, L. Corrigan, Predicting market movement direction for Bitcoin: A comparison of time series modeling methods, Comput. Electr. Eng. 89 (2021) http://dx.doi.org/10.1016/j.compeleceng.2020.106905.
[48] R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 161–168, http://dx.doi.org/10.1145/1143844.1143865.
[49] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction, 2nd ed., Springer, 2009, http://www-stat.stanford.edu/tibs/ElemStatLearn/.
[50] I. Guyon, B. Boser, V. Vapnik, Automatic capacity tuning of very large VC-dimension classifiers, in: NIPS. Morgan Kaufmann, 1992, pp. 147–155.
[51] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1999.
[52] T. Wiesel, Receptive fields and functional architecture of monkey striate cortex, J. Physiol. (1968) 215–243.
[53] K. Fukushima, Neocognition: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybernet..
[54] M. Gudelek, S. Boluk, A. Ozbayoglu, A deep learning based stock trading model with 2-D CNN trend detection, in: IEEE Symposium Series on Computational Intelligence, SSCI, 2017, pp. 1–8.
[55] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, WaveNet: A generative model for raw audio. Google DeepMind, London, UK, 2016, pp. 1–15, http://arxiv.org/abs/1609.03499.
[56] P. Lamothe-Fernandez, D. Alaminos, P. Lamothe-López, M.A. Fernández-Gámez, Deep learning methods for modeling Bitcoin price, Mathematics 1245 (8) (2020) http://dx.doi.org/10.3390/math8081245.
[57] M. Mamun, G. Uddin, Mt. Suleman, Geopolitical risk, uncertainty and Bitcoin investment, Physica A 540 (2020).
[58] Z. Chen, C. Li, W. Sun, Bitcoin price prediction using machine learning: An approach to sample dimension engineering, J. Comput. Appl. Math. 365 (2020).
[59] M. Liu, G. Lia, J. Lia, X. Zhua, Y. Yaoa, Forecasting the price of Bitcoin using deep learning, Finance Res. Lett. (2020) 1544–6123.
[60] R. Jana, I. Ghosh, D. Das, A differential evolution based regression framework for forecasting Bitcoin price, Ann. Oper. Res. (2021) http://dx.doi.org/10.1007/s10479-021-04000-8.
[61] J.M. Chatterjee, A. AbdAllah, A Bayesian regularized neural network for analyzing Bitcoin trends, IEEE Access 9 (2021).
[62] T.N. Nguyen, N. Hamdi, K. Cengiz, Global cryptocurrency trend prediction using social media, Inf. Process. Manage. 58 (2021).