BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 2
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0020

Cryptocurrency Price Prediction Using Enhanced PSO with Extreme Gradient Boosting Algorithm

Vibha Srivastava, Vijay Kumar Dwivedi, Ashutosh Kumar Singh

Department of Computer Science and Engineering, United College of Engineering and Research, Naini, Prayagraj, 211010, Uttar Pradesh, India.
E-mails: vibha02.02@gmail.com vijay.kr.dwivedi@gmail.com ashuit89@gmail.com

***Abstract**: Due to the highly volatile tendency of Bitcoin, there is a necessity for a better price prediction model. Only a few researchers have focused on the feasibility to apply various modelling approaches. These approaches may prone to have low convergence issues in outcomes and acquire high computation time. Hence a model is put forward based on machine learning techniques using regression algorithm and Particle Swarm Optimization with XGBoost algorithm, for more precise prediction outcomes of three cryptocurrencies; Bitcoin, Dogecoin, and Ethereum. The approach uses time series that consists of daily price information of cryptocurrencies. In this paper, the XGBoost algorithm is incorporated with an enhanced PSO method to tune the optimal hyper-parameters to yield out better prediction output rate. The comparative assessment delineated that the proposed method shows less root mean squared error, mean absolute error and mean squared error values. In this aspect, the proposed model stands predominant in showing high efficiency of prediction rate.*

***Keywords**: PSO; XGBoost; cryptocurrency; price prediction; Regression Algorithm.*

1. Introduction

The exponential progress in access to the Internet has triggered new techniques and technologies in the real world. Cryptocurrency stands as one of the emerging Internet technology utilization to be currency, overlapping the traditional monetary system [1]. This cryptocurrency term refers to the virtual currencies that work as exchange models or assets transformation digitally. The cryptocurrency has been introduced in the year 2009, referred to as Bitcoin by scientist Satoshi Nakamoto [2].

The assessment of empirical researchers and theoretical studies have propounded that cryptocurrencies' price dynamics have been impacted by various latent parameters [3]. Those key parameters have not been yet clearly defined and determined. The vast group of authors are inclined to believe that those fundamental parameters do not have significant impacts on the rate of a cryptocurrency, rather than prices of cryptocurrency were determined through the demand-supply ratio. In recent research, complex system theory methods are utilized and described the possibility to construct crash and critical phenomena indicators in cryptocurrency markets and volatile stock markets [4]. Results have turned out to be consistent compared to different empirical researchers employing statistical methods. In spite of developing interest, cryptocurrencies integration and acceptance within global financial markets, there is only very limited researcher to model out the volatility dynamics and price prediction of cryptocurrency. Such kind of unstable variations seems to be tedious for user prediction [5]. Various factors might affect the cryptocurrencies' prices, for many years, wherein it may include external and internal factors. The parameters that are associated with crypto markets such as volatility, market beta and trading volume could be obtained to be one of the significant aspects that determine the prices of cryptocurrency (from year 2010 to year 2020). To generate an accurate price prediction method for the sort of complex issues like their performance, convergence rate seems to be highly challenging [6].

Another issue to be addressed is the convergence speed of the prediction model and cryptocurrencies' recognition to be financial instruments. The inspiration for studies addressing these issues is the acceptance speed and cryptocurrency recognition to be financial instruments [7]. The increased Machine Learning (ML) techniques utilization for time series and cost prediction issues in seeking more precise outcomes of the prediction phase is another inspiration to focus on such a study. This inspiration must focus on synthesizing the collective knowledge from prior years regarding the development and operation of cryptocurrency and highlighting the prominent methods adopted for cryptocurrency price prediction. Even though the training of random particles or dataset trading features is held, using conventional approaches, the lacking to tune the parameters prior to handling the local particle solutions exists in many studies. The global and local optima of the features need to be dealt with prior to the prediction phases of price. Therefore, the study is put forward to solve these issues.

The major contributions of the paper are stated below:

- To Apply Dynamic inertia weight approach regression model, Enhanced PSO XGBoost method, propound for cryptocurrency price prediction, performs precise price prediction of cryptocurrencies.
- To determine the global and local optimal solutions of particles, using PSO algorithm with dynamic inertia weight updated function, perform hyperparameters tuning of those weight updated parameters, using XGBoost algorithm, to explicate the global optimal solution, aiding better price prediction outputs of cryptocurrencies.
- The training of global weighted optimal features, enables to predict the more accurate price estimation of cryptocurrencies, with fast convergence function.

The remainder organization of the paper is as follows. Some existing works related to cryptocurrency features and approaches for price prediction are enumerated in Section 2. Section 3 elucidates the research methodology of the work. Section 4 discusses the results analysis of the study. Section 5 propounds the conclusive part of the study.

2. Related work

Existing research, discussing the different approaches of cryptocurrency price forecasting models and features in the market are enumerated in this section.

2.1. Features of cryptocurrencies and forecasting of cryptocurrencies prices

A cryptocurrency is defined as a network-based medium for electronic exchange, wherein the records are secure through stronger cryptographic algorithms including Message Digest-5 (MD-5) and Secure Hash Algorithm. It generally utilizes blockchain technology in making secure transactions, immutable, traceable and transparent transactions. Due to those attributes, it has gained popularity specifically in the financial streams. Cryptocurrencies are characterized by the count of transactions and are majorly volatile [8, 27]. The cryptocurrencies, as the name implies, is referred to because of the usage of encryption method for regulation of coin transfer and coin creation. It seems to be essential to make clear the financial and social factors determining Bitcoin prices. Hence, this could be beneficial to understand the nation's economy. Different cryptocurrencies and Bitcoin had not moved out well because of avoiding the financial system and in maximization of impossibility to permit cash movement and illegal activities fights. This consists of Chinese government's decisions in getting rid of cryptocurrency (Bitcoin) in year 2013, Mt.Gox bankruptcy, one of Bitcoin trading heads and to yield legitimacy after vote of Brexit [9]. This scenario and various other conflicts had resulted to the demand to study the digital currencies. One such study similarly explicates prediction model for price fluctuation of multiple cryptocurrencies.

Hence in the recent period, different technique has been proposed by different researcher in the modelling and prediction of cryptocurrencies prices. By the way, different researchers also have been involved to assess the crypto market volatility. The interest level in cryptocurrencies similarly increases, after the high crash of cryptocurrencies (referred as Bitcoin crashes) at starting month of 2018. In case of cryptocurrency trading, the observers of domain and other traders ought to perform analytics on cryptocurrencies (primarily Bitcoin) and ought to estimate the price of cryptocurrency. The terms forecasting and prediction of prices have been used generally in same approach that refers to prediction task, a price estimate on the basis on past prices history and various explanatory costs. The “prediction” term, seems to be high general since it means the current price prediction or future price prediction while this forecasting have been utilized for referring to make future price estimation or trends estimates [10].

2.2. Machine learning approach of forecasting cryptocurrency prices

The major variation between classical modelling and ML is the typical difference wherein this ML algorithm does data interpretation themselves; hence, there is not a necessity in performing initial decomposition. Depending on the objectives of the analysis, the ML algorithm builds logic modeling based on the data available. This could prevent the lengthy and complex pre-model statistical testing stages of different hypothesis. The major objective of the study in determining the capability of ML techniques, to assess effectively time series cryptocurrency data (vector and scalar). One such study also aids in determining time correlation and data patterns, forming the base data for qualitative data forecasts. This significant ML characteristic is that this technique utilized in searching templates within data that does not infer a relationship type, statistical properties and prior data structure [11]. In this paradigm of ML, different count of effective methods, algorithms and approaches have been developed including Support Vector Method (SVM), kernel methods, Artificial Neural Networks (ANN), Gradient Boosting (GBoost), Deep learning approaches, Regression classification ensembles Random Forest (RF) techniques, etc. The implementation of ML approaches has been utilized in analyzing and forecasting time series specific data of cryptocurrency.

Some researchers are capable of anticipating various degrees and different fluctuations of prices (Bitcoin). The researchers have also delineated that best outcomes are also achieved using NN-based algorithms. One method is a reinforcement learning technique implemented to beat normalized by and in holding out strategies for prices prediction of nearly twelve cryptocurrencies across one-year period [12]. Similarly, the daily cryptocurrency market trends are determined that assess the features, associated to cryptocurrency price using a regression model. The dataset used in the study comprises of nine features related to price of cryptocurrency recorded on daily basis across a six-month period. The daily prices change of cryptocurrency is performed through ML algorithms. One such study that employs a multivariate linear regression model is utilized for predicting the highest price prediction and low price estimation [13]. This phase of prediction could offer benefits to cryptocurrency traders and stockbrokers uplifting hands within the market. It facilitates miners to mine Bitcoins with good profits. The model training in ML could be deployed in real-time and results perform well than Linear Regression (LR) and SVM specifically for large volumes of datasets [14].

Likewise, the relations between Ethereum prices and inherent Ethereum blockchain information have been investigated in another study. The impact of blockchain how it concerns available coins on the global market and its relation with prices are also investigated. The key findings illustrate blockchain information (Ethereum-specific) and how it is related to the price prediction phase [15]. Owing to this another model Multivariate Multistep output LSTM (MMLSTM) model is applied to generate stock close price value prediction for one week for a firm [16]. Hence this ML approach could result in profitable strategies within the market of cryptocurrency and highly realistic features of the market are assumed if market circumstances have fluctuations [17].

2.3. Research gaps

- Generally, research results in price prediction rely on different types of datasets, having trading features in various time periods. Hence, the outcomes could not compare fairly to attain a conclusive statement to recommend a single prediction method across another one. A research to address this issue must be employed [18].
- Few sorts of confidence or trust score measure ought to be accounted for missing factors, uncertainty aiding the accuracy of price prediction, such that on

entire performance model is recommended [19]. Non-linear techniques and tests have not been explored thoroughly for application in time series of cryptocurrencies. It ought to be investigated further in capturing non-linear dependencies on those explanatory variables, due to the issue of low convergence rate [20].

- It needs to be noted wherein a minimal dataset is used with only a few lag values of time series of digital coins closing prices. Additionally, in a few studies, a variety of oscillators, and indices that moves an average of various time periods and types can be used in accounting to trend dynamics. Due to low feature values, it may evolve around only locally optimal solutions [21].
- In some studies, the prediction of cryptocurrency prices does not attempt in exploiting occurrences of different cryptocurrency prices on different types of exchanges and this consideration would open the approach towards higher returns on investment [22]. In those prediction models, the intraday price fluctuations are ignored in trading features instead average daily prices fluctuations are assumed. The highest and lowest attained price fluctuations need to be considered [23].

3. Research methodology

The input datasets such as Bitcoin dataset, Dogecoin, and Ethereum dataset are loaded into pre-processing phase, wherein the feature scaling and removing of missing values are performed.

The scaling of the feature enables fixing the common range of all increases and decreases in the statistical close prices and open prices of the cryptocurrencies. The missing values within the dataset are removed from the model and the fixed-scaled values are fed to the regression phase. In this regression phase, the optimization algorithm named the enhanced PSO method is put forward to handle the local and global position and velocity of the particles [28]. This seems to be different from the normal PSO model, wherein enhanced PSO could adjust inertia weight co-efficient dynamically to enhance the convergence speed of the algorithm in price prediction in terms of the iteration count. The inertia weight coefficient is assigned to those local and global learning objects. Hence, this global particle, an optimal solution, is fed as super parameters to the XGBoost algorithm. The tuning of hyperparameters is updated in this algorithm facilitates to bring out the best accurate price prediction of cryptocurrencies with maximum iterations in learning and rapid convergence tendency. For every iteration, the global and local optimal particle values are updated. The best parameters yield better results with high convergence obtained as hyperparameters in this tuning process. Hence, optimal solutions are fetched and segregated into the training phase and testing phase. The optimal solutions represent the best accurate price prediction rate of each dataset individually. In accordance with the performance metrics, the efficiency of the proposed cryptocurrency price prediction model is evaluated through different metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The whole process is shown in Fig. 1.

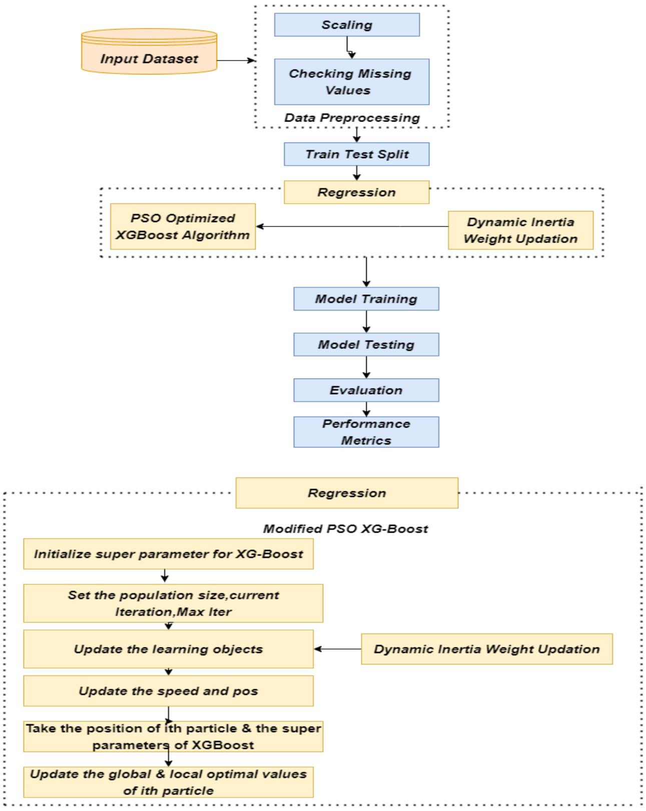

Fig. 1. Flow diagram of proposed model

3.1. Dataset description

The BTC.csv, Doge-USD.csv, and ETH_1H.csv files are the datasets imported for this study. The numerical input for the respective study used in each of the models is represented with the data, price open time along with the volume BTC and the volume USD. Whereas the output of the numerical values forms, the respective model is the closing price of the Bitcoin. It comprises one-hour historical trading data features from 2017 to 2021 trading features. The dataset consists of seven columns such as the timestamp of data, dates, symbol, the opening price of a cryptocurrency, closing prices of digital coins, adjacent closing price, and volume of a cryptocurrency. The total count of records for Bitcoin is 2991, Dogecoin represents 2544 and Ethereum consists of a total of 2160 records.

3.2. PSO Algorithm with dynamic inertia weight coefficient adjustment – optimization of best values

PSO Algorithm is a population based stochastic optimization technique, explicated as effective and efficient global optimizer. In this proposed method, the standard PSO Algorithm performance is enhanced through utilizing dynamic inertia-weight, which lowers in accordance to increase in iterative generation.

PSO Algorithm is utilized to determine the potential optimal solution within search space. In this phase, it aids to come out with better solution for prediction phase. Every particle in PSO represents potential solution in this search space of cryptocurrency price variations with open price, close price value in dataset. In this d-dimensional space, the velocity vector (vv) of i-th particle could be denoted as $\mathrm{vv}_i = (\mathrm{vv}_{i1}, \mathrm{vv}_{i2}, \mathrm{vv}_{i3}, \ldots, \mathrm{vv}_{iD})$ and position vector (pv) of this particle points to $\mathrm{pv}_i = (\mathrm{pv}_{i1}, \mathrm{pv}_{i2}, \mathrm{pv}_{i3}, \ldots, \mathrm{pv}_{iD})$. The random-values (rv) that are uniformly distributed within range [0, 1] are denoted by rv_1& rv_2.

Algorithm 1. Standard Particle Swam Optimisation Algorithm

$Input$: Initialize population

$Output$: Optimal value of weight

Step 1. $--\rightarrow$ **Initialisepopulation**

Step 2. $--\rightarrow$ $\mathbf{for}\ t = 1$: maximumgeneration

Step 3. $--\rightarrow$ $\mathbf{for}\ i = 1$: populationsize

Step 4. $--\rightarrow \mathbf{iff}\left(\mathrm{pv}_{i,D}(\boldsymbol{t})\right) < f(w_{\mathrm{g}}(\boldsymbol{t})$ $\mathbf{then}$ $w_{\mathrm{g}}(\boldsymbol{t}) = \mathrm{pv}_{i,D}(\boldsymbol{t})$

Step 5. $--\rightarrow \boldsymbol{f}\left(w_{\mathrm{g}}(\boldsymbol{t})\right) = \mathbf{min}(\boldsymbol{f}(w_i(\boldsymbol{t})))$

Step 6. $--\rightarrow$ **end**

Step 7. $--\rightarrow \mathbf{for}\ \boldsymbol{d} = \mathbf{1}$: **dimension**

Step 8. $--\rightarrow \mathrm{vv}_{i,D}(\boldsymbol{t}+\mathbf{1}) = \omega \mathrm{vv}_{i,D}(t) + a_1 \mathrm{rv}_1\left(w_i - \mathrm{pv}_{i,D}(t)\right) +$
$a_2 \mathrm{rv}_2\left(w_{\mathrm{g}} - -\mathrm{pv}_{i,D}(t)\right)$

Step 9. $--\rightarrow \mathrm{pv}_{i,D}(t+1) = \mathrm{pv}_{i,D}(t) + \mathrm{vv}_{i,D}(t+1)$ (2)

Step 10. $--\rightarrow$ **if** $\mathrm{vv}_{i,D}(\boldsymbol{t}+\mathbf{1}) > \mathbf{vv_{max}}$ **then** $\mathrm{vv}_{i,D}(\boldsymbol{t}+\mathbf{1}) = \mathbf{vv_{max}}$

Step 11. $--\rightarrow$ **else if** $\mathrm{vv}_{i,D}(\boldsymbol{t}+\mathbf{1}) < \mathbf{vv_{min}}$ **then** $\mathrm{vv}_{i,D}(\boldsymbol{t}+\mathbf{1}) = \mathbf{vv_{min}}$

Step 12. $--\rightarrow$ **end**

Step 13. $--\rightarrow$ **if** $\mathrm{pv}_i(t+1) > \mathbf{pv_{max}}$ **then** $\mathrm{pv}_i(t+1) = \mathbf{pv_{max}}$

Step 14. $--\rightarrow$ **else if** $\mathrm{pv}_i(t+1) > \mathbf{pv_{min}}$ **then** $\mathrm{pv}_i(t+1) = \mathbf{pv_{min}}$

Step 15. $--\rightarrow$ **end**

Step 16. $--\rightarrow$ **end**

Step 17. $--\rightarrow$ **end**

Step 18. $--\rightarrow$ **end**

Once the initialization of dataset particles is made randomly, i-th particle's velocity vector and then position vector representations are updated by the next equations:

(1) $$\mathrm{vv}_i(t+1) = \omega \mathrm{vv}_i(t) + a_1 \mathrm{rv}_1(w_i - \mathrm{pv}_i(t)) + a_2 \mathrm{rv}_2(w_{\mathrm{g}} - \mathrm{pv}_i(t)),$$

(2) $$\mathrm{pv}_i(t+1) = \mathrm{pv}_i(t) + \mathrm{vv}_i(t+1).$$

In the above equations, w represents inertia weight parameter.

The weight parameter is enhanced with PSO Algorithm to optimize the solutions (particles). This parameter is utilized in controlling the impact of previous velocity vector on new particle velocity. The constant parameters are denoted by a_1, a_2. Similarly, the weight added with particle solutions also consist of best previous i-th particle (feature) position represented by w_i. In the current generation, all the particles previous position vector is pointed as w_{g}.

Algorithm-2. Enhanced Particle Swarm Optimisation Algorithm

$Input$: Initialisepopulation

Step 1. $--\rightarrow$ $\boldsymbol{Intialization}$

Step 2. $--\rightarrow$ $Begin$

Step 3. $---\rightarrow \mathrm{cv}_1 = \mathrm{constantvariable}$
Step 3.1. $---\rightarrow \mathrm{cv}_2 = \mathrm{constantvariables}$
Step 3.2. $---\rightarrow m_{\max} = \mathrm{constantvariable}$
Step 3.3. $---\rightarrow \mathrm{ip}_i^m \rightarrow \mathrm{Positionofagroupofparticles}$
Step 3.4. $---\rightarrow v_i^m \rightarrow \mathrm{Velocitiesofgroupofparticles}$
Step 4. $---\rightarrow$ **Optimizations**
Step 5. $---\rightarrow$ **Evaluate**
Step 6. $---\rightarrow \mathrm{Fitnessofparticles} = \mathrm{pf}_m^i, \mathrm{findforeachparticles}$
Step 7. $---\rightarrow \mathrm{comparingeachfitnessparticlewithbestoneBFbest}_i$
Step 8. $---\rightarrow$ **if**
Step 9. $---\rightarrow \mathrm{pf}_m^i \leq \mathrm{pf}_{\mathrm{best}}^i$
Step 10. $---\rightarrow$ **then**
Step 11. $---\rightarrow \mathrm{newpositionofithparticleisbetterthanBFbest}_i$
Step 12. $---\rightarrow$ **then**
Step 13. $---\rightarrow$ **set**
Step 14. $---\rightarrow \mathrm{pf}_m^i = \mathrm{pf}_{\mathrm{best}}^i$
Step 15. $---\rightarrow \mathrm{BFbest}_i^m = \mathrm{ip}_i^m$
Step 16. $---\rightarrow \mathrm{comparepf}_m^i(\mathrm{individuals})\mathrm{withGPbest}^m$
Step 17. $---\rightarrow \mathrm{pf}_m^i \leq \mathrm{pf}_{\mathrm{best}}^{\mathrm{gb}}$
Step 18. $---\rightarrow \mathrm{newpositionofithparticleisbetterthanGPbest}^m$
Step 19. $---\rightarrow$ **then**
Step 20. $---\rightarrow$ **set**
Step 21. $---\rightarrow \mathrm{pf}_{\mathrm{best}}^{\mathrm{gb}} = \mathrm{pf}_m^i$
Step 22. $---\rightarrow \mathrm{GPbest}^m = \mathrm{ip}_i^m$
Step 23. $---\rightarrow \mathrm{Calculatetheinertiaweightusingbelowequation}$
Step 24. $---\rightarrow \omega = \omega_{\min}\left(\frac{\mathrm{sequential}_{\max}-\mathrm{sequential}}{\mathrm{sequential}_{\min}}\right)^n \times (\omega_{\max} - \omega_{\min}) \quad (5)$
Step 25. $---\rightarrow \mathrm{Updateallparticlevelocitiesaccordingto}$
Step 26. $---\rightarrow v_i^{m+1} = w \times v_i^m + \mathrm{cv}_1 \times \mathrm{rv}_1(\mathrm{BFbest}_i^m - \mathrm{ip}_i^m) +$
$\mathrm{cv}_2 \times \mathrm{rv}_2(\mathrm{GPbest}^m - ip_i^m) \quad (4)$
Step 27. $---\rightarrow \mathrm{Updateallparticlepositionsaccordingtobelowequation}$
Step 28. $---\rightarrow \mathrm{ip}_i^{m+1} = \mathrm{ip}_i^m + v_{im+1}^{m+1} \quad (3)$
Step 29. $---\rightarrow \mathrm{Incrementm}$
Step 30. $---\rightarrow$ repeat Steps 6 to Step 23 until asufficient good fit inessora maximum number of iteration sa rereached.
$Output$: Optimal value of weight

PSO Algorithm1 with inertia weight allocation could obtain least iterations count in order to determine global optimal solution in search space and it lies within range of [0.90, 1.20]. Global optimum defined as optimized solution for any optimization issue. This inertia weight adaptation mechanism in enhanced PSO explored to enhance global optimization capability of PSO technique. It employs fixed parameters through assessing constant parameter that concerns algorithm's direction rather to define on initialization phase manually. This sort of inertia weight adaptation mechanism dynamically applied rather assigning constant inertia weight parameter, for improvising global optimization capability of PSO Algorithm.

To prevent the update rate of each particle, and not to adapt towards every optimization process stage, the non-linear dynamic inertia weight allocation method is employed based on similarity concepts. Hence it facilitates the search process to be highly robust additionally. This dynamic weight adaption method to every particle rectifies the issue of gaining a locally optimal solution. It also proceeds to do an optimization search at similar dynamic time and it maximizes the particle's population diversity within iteration. In this PSO, particles are commonly referred to for multiple randomized candidate solutions and have been maintained inside the search space of the problem, wherein every particle denotes a solution to the optimization issue. Every particle has been analyzed through fitness function, in figuring out if the particle is the best solution for a problem or not it is not. After this assessment, individual particle flies through search space with random velocity by integrating the current solution location and best solution locations.

Let in Algorithm2, the swarm's size be *D* and every particle within dimension vector be represented by *i*. The current position of particle is denoted by $\overrightarrow{\mathrm{cp}}_{\mathrm{l}}$ and the velocity of particle denoted by $\vec{v}_{\mathrm{l}}$. The best value of particle as in the algorithm is represented by $\overrightarrow{\mathrm{bv}}_{\mathrm{l}}$. This enhanced PSO Algorithm is composed to adjust the velocity of every particle and particle's position to current best location and global best solutions. At every step, the present $\overrightarrow{\mathrm{cp}}_{\mathrm{l}}$ position has been updated through velocity and assessed to be solution of problem. If the particle determines a pattern, that seems better than others, of previous ones, then it has been recorded in vector $\overrightarrow{\mathrm{bv}}_{\mathrm{l}}$. The resultant best fitness function value has been observed as BFbest_i to compare with next iterations. Like this similarly, PSO Aalgorithms keeps on determining better particle positions, and keep updating both $\overrightarrow{\mathrm{bv}}_{\mathrm{l}}$ and BFbest_i.

The individual particles' position ip_i at *m*+1 iteration gets changed in according to the next equation,

(3) $$\mathrm{ip}_i^{m+1} = \mathrm{ip}_i^m + v_{im+1}^{m+1}.$$

The position of particle is adjusted through velocity of particle, computed by the next equation,

(4) $$v_i^{m+1} = w \times v_i^m + \mathrm{cv}_1 \times \mathrm{rv}_1(\mathrm{BFbest}_i^m - \mathrm{ip}_i^m) + \mathrm{cv}_2 \times \mathrm{rv}_2(\mathrm{GPbest}^m - \mathrm{ip}_i^m),$$

wherein $i = 1, 2, \dots, n$ and iteration index is denoted by *m*; ip_i^m, v_i^m represent the position of particle (*i*) and velocity of particle (*i*) at specific *m*-th iteration. Then in Algorithm 2, the best particle's position is assigned to be BFbest_i^m at iteration *m*. In entire swarm, global best particle's solution is represented and assigned to variable GPbest^m until *m* iteration. After finding GPbest^m value, the inertia weight parameter is assigned, where this factor adjusts particle's velocity dynamically and hence it controls exploitation and exploration of search space. This assigned non-linear decreasing inertia weight parameter *w* is defined by the next equation,

(5) $$w = w_{\min}\left(\frac{\mathrm{sequential}_{\max} - \mathrm{sequential}}{\mathrm{sequential}_{\min}}\right)^n \times (w_{\max} - w_{\min}),$$

wherein maximum iteration count is represented as $\mathrm{sequential}_{\max}$ such that current iteration is sequential. The lower inertia limit values and upper inertia weight limit values are $w_{\min}$ &$w_{\max}$; cv_1 and cv_2 denote the cognitive coefficient and social coefficient parameter. The pre-defined randomized particle's values are rv_1 and rv_2 that lies within range [0, 1]. At every iteration, inertia weight would lower non-linearly from this $w_{\min}$ and $w_{\max}$, where the non-linear index is n. After the computation of inertia weight factor, all velocities and position of particles are updated along with weight values in accordance to the next equations:

(6) $v_i^{m+1} = w \times v_i^m + \mathrm{cv}_1 \times \mathrm{rv}_1(\mathrm{BFbest}_i^m - \mathrm{ip}_i^m) + \mathrm{cv}_2 \times \mathrm{rv}_2(\mathrm{GPbest}^m - \mathrm{ip}_i^m)$,

(7) $\mathrm{ip}_i^{m+1} = \mathrm{ip}_i^m + v_{im+1}^{m+1}$.

The iterations are increased, for every cycle; likewise, this computation is proceeded in every iteration. Hence like this, from the steps of fitness function computation and assigning inertia weight parameter is continued unless better fitness value or maximum iterations count is reached out.

3.3. XGBoost technique for hyperparameters tuning

Global optimal solutions are optimized by PSO, to have a group of satisfactory hyperparameters enhance the regression model performance. Hence, to tune these hyperparameters is significant. However, optimization algorithm of hyperparameters rectifies the manual search dependence on experience and trial and error methods. This tuning process is added to PSO Algorithm to improvise the performance rate of price prediction.

The hyperparameters or super parameters control out regularization or complexity in refining the model. One must employ carefully this optimization process, for tuning the hyperparameters, to yield better prediction outcomes. Since ML is an efficient algorithmic approach, XGBoost possesses various hyperparameters. This enhanced PSO Algorithm could obtain optimal values simultaneously of multiple hyperparameters of XGBoost in an n-dimensional problem space. This enhanced PSO rectifies continuous better searching optimization for XGBoost hyperparameters. The implementation of the XGBoost Algorithm affects the PSO Algorithm, in fitting the values and in enhancing the prediction accuracy rate to a better level.

Algorithm-3. XGBoost Method for Hyperparameter Tuning

$Input$: Initialize sparticles $\mathrm{PV}_i = ((\mathrm{pv}_{i1}, \mathrm{pv}_{i2}, \mathrm{pv}_{i3}, \dots, \mathrm{pv}_{iD})$ with

Step 1. $--\rightarrow$ $\mathrm{pv}_i = (\mathrm{pv}_{i1}, \mathrm{pv}_{i2}, \mathrm{pv}_{i3}, \dots, \mathrm{pv}_{iD}) ----\rightarrow$ position vector

Step 2. $--\rightarrow$ $\mathrm{vv}_i = (\mathrm{vv}_{i1}, \mathrm{vv}_{i2}, \mathrm{vv}_{i3}, \dots, \mathrm{vv}_{iD}) ----\rightarrow$ velocity vector

Step 3. $--\rightarrow$ **for** $i = 1; i \le N; i = i + 1$ do

Step 4. $--\rightarrow$ computethelocaldensityβ_i , distanceμ_i ; $\delta(i)$& $\rho(i)-\rightarrow$ β_i , μ_i,-.,-.,-.,-.,-.

Step 5. $--\rightarrow$ Chooseparticleswithhighβ_i&relativelyhigh $\mu_i$asthecenteracctoϑ$_i = \beta_i * \mu_i(8)$

Step 6. $--\rightarrow$ assign remaining particle sand get $S_g$subgroups

Step 7. $--\rightarrow$ ***initialize***

Step 8. $--\rightarrow$ the XG Boostwithinstances nodesset Lontraining data, the hyper parameter$-\rightarrow$ current optimal value

Step 9. $--\rightarrow$ **for** $m = 1; m \le n; m = m + 1$ do

Step 10. $--\rightarrow$ $\mathrm{gain} \rightarrow 0, G = \sum_{i \in L} \mathrm{gain}_i \rightarrow 0, H = \sum_{i \in L} \mathrm{gain}_i \rightarrow 0$

Step 11. $--\rightarrow$ **for** jinsorted(Lbypv$_{i1}$)do

Step 12. $--\rightarrow$ $G_0 \rightarrow G_0 + g_0, H_0 \rightarrow H_0 + h_0$;

Step 13. $--\rightarrow$ $G_p \rightarrow G + G_0, H_0 \rightarrow H + H_0$;

Step 14. $--\rightarrow$ $\max_{\mathrm{score}}\left(\mathrm{score}, \frac{G_0^2}{H_0} + \frac{G_p^2}{H_p} + \frac{G^2}{H+\lambda}\right)$

Step 15. $--\rightarrow$ Updateparticlesstate(BFbest_i^m, GPbest^m)refertolossfunction

Step 16. $--\rightarrow$ for $d = 1; d \leq D; d = d + 1$ do
Step 17. $--\rightarrow$ if particleislocaloptimal then
Step 18. $--\rightarrow$ $v_i^m = w \times v_i^m + \mathrm{cv}_1 \times \mathrm{rv}_1(\mathrm{BFbest}_i^m - \mathrm{ip}_i^m) + \mathrm{cv}_2 \times \mathrm{rv}_2(\frac{1}{d}\sum_{d=1}\mathrm{GPbest}^m - \mathrm{ip}_i^m)$ (9)
Step 19. $--\rightarrow$ $\mathrm{ip}_i^m = \mathrm{ip}_i^m + v_i^m$ (10)
Step 20. $--\rightarrow$ else
Step 21. $--\rightarrow$ $v_i^m = w \times v_i^m + \mathrm{cv}_1 \times \mathrm{rv}_1(\mathrm{BFbest}_i^m - \mathrm{ip}_i^m) + \mathrm{cv}_2 \times \mathrm{rv}_2(\mathrm{GPbest}^m - \mathrm{ip}_i^m)$
Step 22. $--\rightarrow$ $\mathrm{ip}_i^m = \mathrm{ip}_i^m + v_i^m$ (10)
$Output$: Optimal value of hyperparameters

The position vector and velocity vector of particles are initialized. The local density of particles is computed by the equation

(8) $$\delta(i) \& \rho(i) -\rightarrow \beta_i, \mu_i.$$

In accordance with the determined value by the particle's position, the XGBoost hyperparameters are assigned and verification data is introduced in the prediction phase. Then the loss functionality on the dataset determines the particle's fitness function value. The optimal current value is the XGBoost parameter having instances node set upon training data (Hyperparameter). The particles are segregated into ordinary particles and optimal particles, according to fitness function value. Algorithm 3 updates the information of hyperparameters of corresponding particles and it checks out if the condition of termination is attained. If it is reached, the optimal hyperparameter optimal value is gained. Those parameters yield better prediction outcomes. The optimized hyperparameters are used in constructing the effective precise price prediction model for cryptocurrencies.

The XGBoost model is built with corresponding hyperparameters obtained by the current best solution. The loss function updates the dataset prediction and fitness function value. The global optimal particles $\mathrm{GPbest}^{\mathrm{m}}$ and local optimal particles $\mathrm{BFbest}_{\mathrm{i}}^{\mathrm{m}}$ are determined in accordance with population diversity and particle fitness function. Then positions of local and global optimal solutions are updated as in the next equations:

(9) $$v_i^m = w \times v_i^m + \mathrm{cv}_1 \times \mathrm{rv}_1(\mathrm{BFbest}_i^m - \mathrm{ip}_i^m) + \mathrm{cv}_2 \times \mathrm{rv}_2\left(\frac{1}{d}\sum_{d=1}\mathrm{GPbest}^m - \mathrm{ip}_i^m\right),$$

(10) $$\mathrm{ip}_i^m = \mathrm{ip}_i^m + v_i^m.$$

If the maximum number of iterations are not reached, then optimal values of hyper parameters are returned, otherwise the hyper parameters of the XGBoost model are again identified using best particle solution (PSO Inertia Weight optimization) by PSO. Once the iterations are reached, hence the optimal hyperparameters are obtained that build out the XGBoost model.

The parameters are tuned for all the three-dataset obtained from the Bitcoin, Dogecoin and for the Ethereum coin dataset.

Hyperparameters Tuning for each dataset

Each of the parameters of PSO Algorithm are tuned using the proposed novelty, Dynamic weight updation in the PSO Algorithm, such as inertial weight, C1, C2, popsize, iteration, gbest and NParticle. These are depicted in the table given in Table 1 by showing the list of values and the best parameters obtained.

Table 1. Proposed PSO Algorithm

Parameters	List of values	Best parameters
inertia weight *w*	[0.1, 0.2, 0.3, 0.4, 0.5, 0.8]	0.8
c1(acceleration constants)	[1, 2, 6]	2
c2(acceleration constants)	[1, 2, 6]	2
popsize	[30, 50]	30
iteration	500	500
gbestfitness	0	0
gbest	random	set the parameters of the xgboost
NParticle	[40, 50]	40
MaxIters	1000	1000
Ndim	[10, 20]	10

Initially, the Bitcoin datasets are taken for the process of tuning using the XGBoost algorithm. These data comprise some of the parameters such as eta, Max_depth, min_child_weight, subsample, and gamma. These have been tested using the various list of values and the best parameter is obtained for each of the parameters. These values have been then taken for evaluation using the performance metrics, such as MSE, RMSE the accuracy rates were obtained as 0.03 and 0.017 respectively. These are represented using the table format below (Table 2), XGBoost classifier for hyperparameter tuning for the Bitcoin dataset.

Table 2. Proposed classifier for Bitcoin

Params	List of values	Best parameters
max_depth	[2, 4, 10, 20, 30, 40, 50]	4
n_estimators	[2, 4, 8, 10, 12, 14, 16, 18, 20]	2
col_sample_tree	[1, 2, 3, 4, 5, 6]	1
subsample	[1, 2, 3, 4, 5, 6]	1
min_child_weight	[0.805, 1.569, 0.415]	1.569
eta	[0.129, 0.084, 0.052]	0.052
gamma	[0.147, 0.476, 0.248]	0.248
Bitcoin MAE	**BitcoinMSE**	**Bitcoin RMSE**
0.071487825	0.030922822	**0.017584886**

Similarly, the XGBoost classifier for hyperparameter tuning for the Dogecoin dataset is represented, using the Table 3. The MSE values and the MSE values of the dataset obtained are 0.00040 and 0.0020, respectively.

Similarly, the XGBoost classifier for hyperparameter tuning for the Dogecoin dataset is represented in the Table 3. The MSE values and the MSE values of the dataset obtained are 0.00040 and 0.0020, respectively.

Table 3. Proposed classifier for Dogecoin

Parameter	List of values	Best parameters
max_depth	[2, 4, 10, 20, 30, 40, 50]	2
n_estimators	[2, 4, 8, 10, 12, 14, 16, 18, 20]	8
col_sample_tree	[1, 2, 3, 4, 5, 6]	5
subsample	[1, 2, 3, 4, 5, 6]	2
min_child_weight	[0.805, 1.569, 0.415]	0.8051
eta	[0.129, 0.084, 0.052]	0.084
gamma	[0.147, 0.476, 0.248]	0.248
Dogecoin MAE	**Dogecoin MSE**	**Dogecoin RMSE**
0.000399667	0.000407408	**0.002018434**

Similarly, the XGBoost classifier for hyperparameter tuning for the Ethereum dataset is shown in the Table 4. The MSE and the RMSE values obtained are 0.037 and 0.019, respectively.

Table 4. Proposed classifier for Ethereum

Params	List of values	Best parameters
max_depth	[2, 4, 10, 20, 30, 40, 50]	4
n_estimators	[2, 4, 8, 10, 12, 14, 16, 18, 20]	2
col_sample_tree	[1, 2, 3, 4, 5, 6]	1
subsample	[1, 2, 3, 4, 5, 6]	1
min_child_weight	[0.805, 1.569, 0.415]	0.8051
eta	[0.129, 0.084, 0.052]	0.129
gamma	[0.147, 0.476, 0.248]	0.147
Ethereum MAE	**Ethereum MSE**	**Ethereum RMSE**
0.006800319	0.037967598	**0.019485276**

4. Results and discussion

4.1. Performance metrics

(11) $$\text{MSE} = \frac{1}{N}\sum_{(x,y)\in D}(y - \text{prediction}(x))^2.$$

Here, N is the number of data in the dataset, D is the dataset, x is the features and y is the label. The prediction function would return the features for x,

(12) $$\text{RMSE} = \frac{\sqrt{\sum_{i=1}^{N}(\text{actual time series observation} - \text{predicted time series observation})^2}}{N}.$$

Here, i represents the variable and N indicates the counts of non-missing data points,

(13) $$\text{MAE} = \frac{\sum_{i=1}^{N}|\text{predicted value} - \text{actual value}|}{N}.$$

Here N indicates the total data point counts.

4.2. Performance evaluation

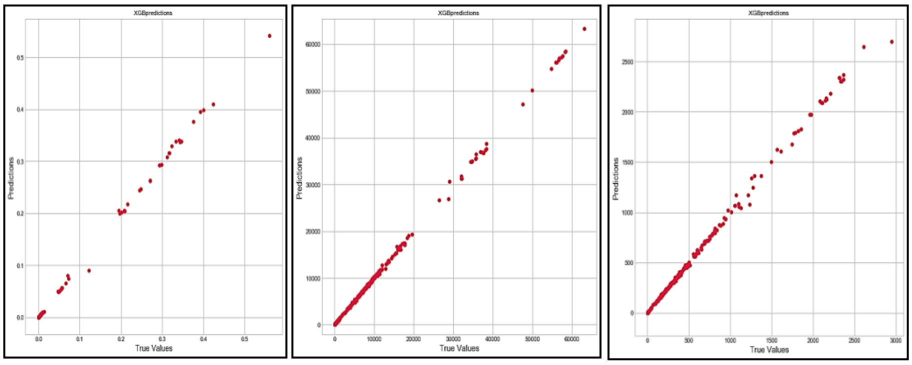

Fig. 2. Predication Data Distribution: (i) Dogecoin; (ii) Bitcoin; (iii) Ethereum

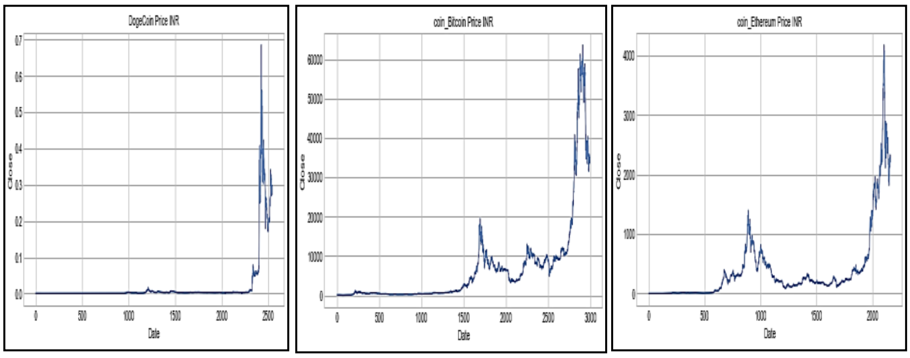

Fig. 3. Closing price prediction Fluctuations rate: (i) Dogecoin; (ii) Bitcoin; (iii) Ethereum

Figs 2 and 3 describe the prediction price values of the Bitcoin, Dogecoin and Ethereum coins with respect to true closing value of the cryptocurrency. The data distribution of predicted cryptocurrency values is presented in the figures representing the prediction value fluctuation across the close price value on different days are clearly defined. From the figures, it can be concluded that proposed prediction model possesses high prediction rate, for all three cryptocurrencies cost closing value.

Table 5. Performance analysis – RMSE and MAE metrics [23]

BITCOIN		
Methods	RMSE	MAE
Existing	369	470
PROPOSED SYSTEM	0.017584886	0.071487825

Table 5 illustrates the performance evaluation of the proposed price prediction model, in terms of determining the RMSE and MAE values. The existing techniques exhibit higher RMSE and MSE values, wherein very low RMSE and MAE values of 0.017584886 and 0.071487825. Hence these outcomes delineated higher performance of the proposed enhanced PSO XGBoost algorithm for accurate price prediction with fewer error metrics [23].

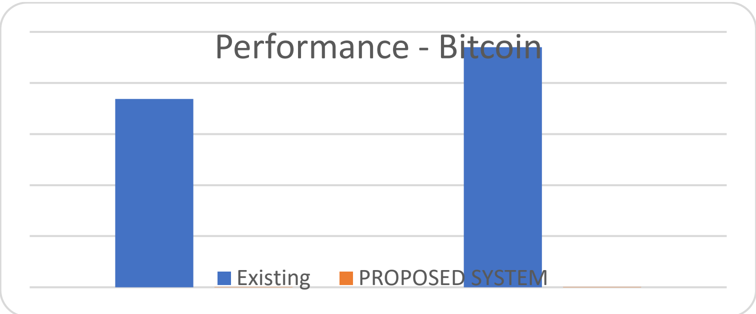

Fig. 4. Performance evaluation of the proposed system (Bitcoin) [23]

Similarly, the Fig. 4 also describes the performance outcomes of the proposed model, in its graphical representation, the performance assessed by low RMSE and MAE values.

4.3. Comparative analysis

Table 6 enumerates the comparative analysis of RMSE metric of proposed prediction regression model with other methods including ForecastX, RNN, DNN method, Multi-variate LSTM, and HOLTS. The existing methods depict different RMSE Values for Bitcoin, Ethereum and Dogecoin. The outcomes of proposed model, exhibited to bring out less RMSE values of 0.017584886 (Bitcoin), 0.002018434 (Dogecoin) and 0.019485276(Ethereum) showing the efficacy of PSO optimization with XGboost hyperparameters tuning contributing for price prediction [24].

Table 6. Comparative analysis table – RMSE metric [24]

Cryptocurrencies/ Methods	RMSE					
	DNN	RNN	FORECASTX	HOLTS	Multivariate LSTM	Proposed system
BITCOIN	0.042	0.067	0.042	0.043	0.044	0.017584886
DOGECOIN	0.123	0.148	0.107	0.102	0.117	0.002018434
ETHEREUM	0.085	0.084	0.059	0.059	0.064	0.019485276

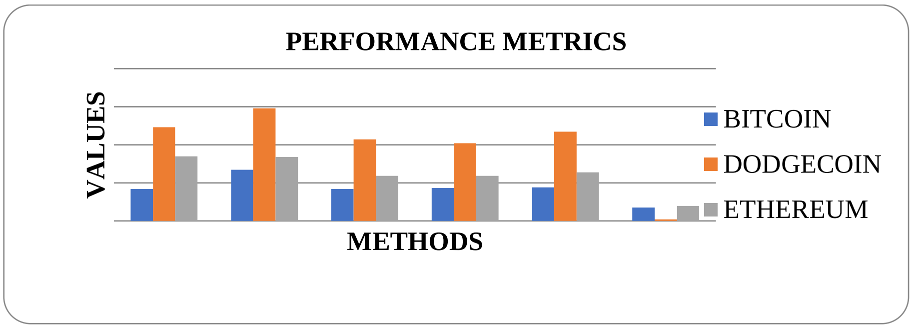

Fig. 5. Comparative assessment of proposed price prediction models of three cryptocurrencies [24]

Fig. 5, illustrates the graphical representation of the comparative assessment of proposed methods for Bitcoin, Ethereum, and Dogecoin. As far from the graphs outcomes, the RMSE values of the proposed system in the last bar enumerates low RMSE values for all the cryptocurrency price predictions showing higher performance in the prediction phase (Table 7).

Table 7. Comparative assessment of the proposed model (MAE, RMSE and MSE) [25]

	Metrics	Bitcoin	Ethereum
Existing system	MAE	313.894	12.949
	MSE	294560	410.01
	RMSE	542.735	20.25
Proposed system	MAE	0.071488	0.006800319
	MSE	0.030923	0.037967598
	RMSE	0.017585	0.019485276

Table 8 shows the comparative analysis of various metrics of MAE, RMSE, and MSE parameters for both existing methods and proposed methods. The existing price prediction model using existing algorithms, generated price prediction with values 313.894(MAE), 294560(MSE) and 542.735 (RMSE) for Bitcoin cryptocurrency and

12.949(MAE), 410.01(MSE) and 20.25(RMSE) for Ethereum [25]. On the other side, these metrics values for the proposed model, for Bitcoin and Ethereum seem to be lesser than 0.04 (low value), indirectly states the prediction rate of cryptocurrency is outperforming other methods.

Table 8. Comparative analysis of proposed method [26]

Cryptocurrencies	Existing system		Proposed system	
	MAE (%)	MSE (%)	MAE (%)	MSE (%)
Bitcoin	3.7	0.5	0.071487825	0.030922822
Ethereum	3	0.3	0.006800319	0.037967598
Dogecoin	3.4	0.3	0.000399667	0.000407408

Similarly, in Table 9, the percentage of MAE and MSE metrics volume are determined individually for Bitcoin and Ethereum for conventional methods and for the proposed model. The percentage of MSE and MAE values of the proposed enhanced PSO XGBoost prediction model generated lesser than 0.07, a low error metrics. This exposed the accurate prediction capability of price estimation of closing prices of cryptocurrencies [26].

Table 9. Parameter values of proposed model

Metrics	Bitcoin	Dogecoin	Ethereum
MAE	0.071487825	0.000399667	0.006800319
MSE	0.030922822	0.000407408	0.037967598
RMSE	0.017584886	0.002018434	0.019485276

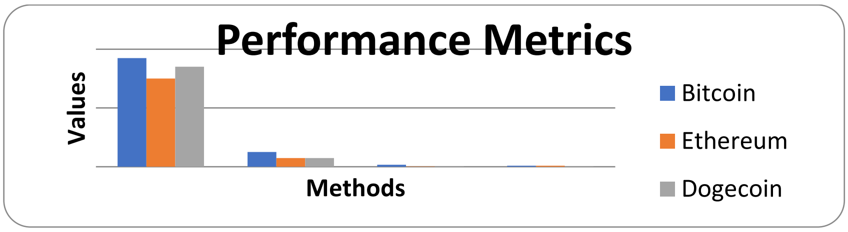

Fig. 6. Comparative assessment of proposed method (MSE and MAE) [26]

Fig. 6 describes the comparative evaluation of performance metrics of cryptocurrency prediction rate of existing methods in terms of percentage with the proposed regression model. From the graphical representation, the proposed method has very low average errors magnitude, MSE and RMSE for all three cryptocurrencies, showing outperforming prediction capability of proposed system.

5. Conclusion

The study explicates cryptocurrency price prediction method, enhanced PSO Algorithm with XGBoost technique. The PSO Algorithm proposed in this study, assigns inertia weight co-efficient dynamically to the global optimal trading feature determined by the PSO approach. The PSO Algorithm applied on pre-processed trading features possesses a stronger ability to pick out the best global optimal solution in continuous search space, facilitating parameter optimization. This global optimal solution is computed with dynamic inertia weight coefficient, to improve the convergence speed and not to stuck on local optimal values. For enhancing the prediction performance, XGBoost technique is employed to tune the optimized hyperparameters, that cope with the best prediction outcomes. The performance of the proposed regression ML technique is assessed through different metrics RMSE, MAE, and MSE. The experimental outcomes of performance analysis of the proposed model, exposed low metrics percentage values of 0.000407408, 0.030922822, 0.037967598 (MSE), 0.000399667, 0.071487825, 0.006800319 (MAE), 0.002018434, 0.017584886, 0.019485276 (RMSE) for Dogecoin, Bitcoin and Ethereum cryptocurrencies. Hence these error metrics, clearly demonstrate the high accuracy performance of proposed regression algorithms, in the closing prices prediction of cryptocurrency.

The future scope of the proposed model could be expanded to different use cases such as forecasting different cryptocurrencies and other stocks.

References

1. Chowdhury, R., M. A. Rahman, M. S. Rahman, M. R. C. Mahdy. Predicting and Forecasting the Price of Constituents and Index of Cryptocurrency Using Machine Learning. – arXiv 2019, arXiv preprint arXiv:1905.08444.
2. Wątorek, M., S. Drożdż, J. Kwapień, L. Minati, P. Oświęcimka, M. Stanuszek. Multiscale Characteristics of the Emerging Global Cryptocurrency Market. – Physics Reports, Vol. **901**, 2021, pp. 1-82.
3. Lucey, B. M., S. A. Vigne, L. Yarovaya, Y. Wang. The Cryptocurrency Uncertainty Index. – Finance Research Letters, Vol. **45**, 2022, 102147.
4. Malladi, R. K., P. L. Dheeriya. Time Series Analysis of Cryptocurrency Returns and Volatilities. – Journal of Economics and Finance, Vol. **45**, 2021, pp. 75-94.
5. Pai, B. A., L. Devareddy, S. Hegde, B. Ramya. A Time Series Cryptocurrency Price Prediction Using Lstm. – In: Emerging Research in Computing, Information, Communication and Applications. Springer, 2022, pp. 653-662.
6. Zhang, Z., H.-N. Dai, J. Zhou, S. K. Mondal, M. M. García, H. Wang. Forecasting Cryptocurrency Price Using Convolutional Neural Networks with Weighted and Attentive Memory Channels. – Expert Systems with Applications, Vol. **183**, 2021, 115378.
7. Fang, F., W. Chung, C. Ventre, M. Basios, L. Kanthan, L. Li et al. Ascertaining Price Formation in Cryptocurrency Markets with Machine Learning. – The European Journal of Finance, 2021, pp. 1-23.
8. Park, J., Y.-S. Seo. Understanding the Association between Cryptocurrency Price Predictive Performance and Input Features. – KIPS Transactions on Software and Data Engineering, Vol. **11**, 2022, pp. 19-28.
9. Ortu, M., N. Uras, C. Conversano, S. Bartolucci, G. Destefanis. On Technical Trading and Social Media Indicators for Cryptocurrency Price Classification through Deep Learning. – Expert Systems with Applications, Vol. **198**, 2022, 116804.
10. Aghashahi, M., S. Bamdad. Analysis of Different Artificial Neural Networks for Bitcoin Price Prediction. – International Journal of Management Science and Engineering Management, 2022, pp. 1-8.
11. Catania, L., S. Grassi. Forecasting Cryptocurrency Volatility. – International Journal of Forecasting, Vol. **38**, 2022, pp. 878-894.
12. Alessandretti, L., A. ElBahrawy, L. M. Aiello, A. Baronchelli. Anticipating Cryptocurrency Prices Using Machine Learning. – Complexity, Vol. **2018**, 2018.
13. Mittal, R., S. Arora, M. Bhatia. Automated Cryptocurrencies Prices Prediction Using Machine Learning. – Division of Computer Engineering, Netaji Subhas Institute of Technology, India, Vol. **8**, 2018, pp. 2229-6956.

14. P o o n g o d i, M., A. S h a r m a, V. V i j a y a k u m a r, V. B h a r d w a j, A. P. S h a r m a, R. I q b a l et al. Prediction of the Price of Ethereum Blockchain Cryptocurrency in an Industrial Finance System. – Computers & Electrical Engineering, Vol. **81**, 2020, 106527.
15. K i m, H. M., G.-W. B o c k, G. L e e. Predicting Ethereum Prices with Machine Learning Based on Blockchain Information. – Expert Systems with Applications, Vol. **184**, 2021, 115480.
16. A a s i, B., S. A. I m t i a z, H. A. Q a d e e r, M. S i n g a r a j a h, R. K a s h e f. Stock Price Prediction Using a Multivariate Multistep LSTM: A Sentiment and Public Engagement Analysis Model. – In: Proc. of IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS'21), 2021, pp. 1-8.
17. S e b a s t i ã o, H., P. G o d i n h o. Forecasting and Trading Cryptocurrencies with Machine Learning under Changing Market Conditions. – Financial Innovation, Vol. **7**, 2021, pp. 1-30.
18. A w o t u n d e, J. B., R. O. O g u n d o k u n, R. G. J i m o h, S. M i s r a, T. O. A r o. Machine Learning Algorithm for Cryptocurrencies Price Prediction. – In: Artificial Intelligence for Cyber Security: Methods, Issues and Possible Horizons or Opportunities. Springer, 2021, pp. 421-447.
19. S a b r y, F., W. L a b d a, A. E r b a d, Q. M a l l u h i. Cryptocurrencies and Artificial Intelligence: Challenges and Opportunities. – IEEE Access, Vol. **8**, 2020, pp. 175840-175858.
20. B a i, C., T. W h i t e, L. X i a o, V. S. S u b r a h m a n i a n, Z. Z h o u. C2P2: A Collective Cryptocurrency up/down Price Prediction Engine. – In: Proc. of IEEE International Conference on Blockchain (Blockchain'19), 2019, pp. 425-430.
21. D e r b e n t s e v, V., A. M a t v i y c h u k, V. N. S o l o v i e v. Forecasting of Cryptocurrency Prices Using Machine Learning. – In: Advanced Studies of Financial Technologies and Cryptocurrency Markets. Springer, 2020, pp. 211-231.
22. H a m a y e l, M. J., A. Y. O w d a. A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. – AI, Vol. **2**, 2021, pp. 477-496.
23. I q b a l, M., M. S. I q b a l, F. H. J a s k a n i, K. I q b a l, A. H a s s a n. Time-Series Prediction of Cryptocurrency Market Using Machine Learning Techniques. – EAI Endorsed Transactions on Creative Technologies, 2021, p. e4.
24. M i l l e r, D., J.-M. K i m. Univariate and Multivariate Machine Learning Forecasting Models on the Price Returns of Cryptocurrencies. – Journal of Risk and Financial Management, Vol. **14**, 2021, No 10, 486.
25. A l a h m a r i, S. A. Using Machine Learning ARIMA to Predict the Price of Cryptocurrencies. – The ISC International Journal of Information Security, Vol. **11**, 2019, pp. 139-144.
26. V l a c h o s G i o v a n o p o u l o s, M. Forecasting Cryptocurrency Price Movements with Predictive Social Media Analytics. 2022.
27. R a d e v a, I., I. P o p c h e v. Blockchain-Enabled Supply-Chain in Crop Production Framework. – Cybernetics and Information Technologies, Vol. **22**, 2022, No 1, pp. 151-170.
28. S i n g h, A. K., S. S r i v a s t a v a. Varna-Based Optimization: A New Method for Solving Global Optimization. – International Journal of Intelligent Systems and Application, Vol. **10**, 2018, pp. 1-15.

Received: 14.12.2022; Second Version: 11.05.2023; Accepted: 19.05.2023