mathematics

Article

Neural Network-Based Bitcoin Pricing Using a New Mutated Climb Monkey Algorithm with TOPSIS Analysis for Sustainable Development

Samuka Mohanty * and Rajashree Dash

Department of Computer Science and Engineering, Siksha 'O' Anusandhan (Deemed to Be) University, Bhubaneswar 751030, India
* Correspondence: mohantysamuka@gmail.com or samukamohanty@soa.ac.in

Abstract: Bitcoin is yet to be assumed as a worthy cryptocurrency and rewarding asset in the global market. As polynomial-based neural networks (PBNNs) are very robust and more accurate in modeling stock price prediction, their advantage in Bitcoin pricing needs to be analyzed. In this study, the robustness of PBNNs, based on Chebyshev (CPBNN) and Legendre (LPBNN), is blended with the proposed algorithm, coined as the mutated climb monkey algorithm (MCMA), to control the estimation of network parameters to accurately predict the one-day-ahead Bitcoin price. The performance was evaluated by a comparative analysis of the testing of both CPBNN and LPBNN with each of the six algorithms under consideration on three different datasets collected within the same time interval. As the use of a few evaluation criteria will not be able to identify an efficient predictor model, this study also proposes the use of a Multi-Criteria Decision-Making (MCDM) framework to rank all models using 15 different evaluation criteria. The ranking of the models clearly indicates that the proposed MCMA algorithm outperforms all other algorithms under study. The convergence plots of the top two models for the datasets also indicate that the PBNN using MCMA for learning predicts better results.

Keywords: Bitcoin price prediction; Chebyshev polynomials; Legendre polynomials; Monkey algorithm; TOPSIS

MSC: 68T07

Citation: Mohanty, S.; Dash, R. Neural Network-Based Bitcoin Pricing Using a New Mutated Climb Monkey Algorithm with TOPSIS Analysis for Sustainable Development. *Mathematics* **2022**, *10*, 4370. https://doi.org/10.3390/math10224370

Academic Editor: Manuel Alberto M. Ferreira

Received: 27 October 2022
Accepted: 17 November 2022
Published: 20 November 2022

1. Introduction

Bitcoin is a well-known cryptocurrency on the global market. It was first formulated and presented in accordance with the structural guidelines specified by Nakamoto [1]. This was the first of its kind to propose the use of a decentralized manager for transactions using cryptography. The network involved in transactions is a chain of nodes called a blockchain. Each time a transaction is initiated, the blocks in the blockchain are modified by using different hashing techniques. The transactions were validated using cryptography. Blockchain mining is used to give away Bitcoins as rewards which can then be used in the exchange of commodities, services, and other monetary benefits. Although Bitcoin is moving forward in a very positive way for some economies, there are still countries like India that are yet to accept it. With a giant ledger such as blockchain, a high level of security imposed by cryptography and growing computing capacities are helping in the success of such cryptofinance [2]. Many researchers have analyzed the hedge and safe-haven properties of Bitcoin. It has been observed that Bitcoin can act as a hedge as well as a safe-haven in some specific horizons but may be a huge diversifier in most cases [3]. In a study, it was proposed that Bitcoin can be a potential safe-haven asset after the market matures [4]. This depends heavily on the liquidity of Bitcoin. Although Bitcoin is being accepted worldwide, its high volatility and high computing cost obstruct its maturity in full swing. Harey [5] sheds light on many myths and facts about Bitcoin, starting from its origin to its future scope. The high volatility of Bitcoin is the greatest hindrance to its growth. Bitcoin is eight times more volatile than the stock market, and hence, its prediction can play a crucial role in paving the way for its success. The Bitcoin market is still in a transient state and needs a huge fueling by regression analysis to efficiently predict the Bitcoin price and its high volatility. The highly volatile feature of Bitcoin prices needs to be analyzed to improve the prediction of its prices in the future. This is an interesting area of research, both in the financial sector and computer science. Boguslavsky et al. [6] analyzed the impact of the COVID-19 pandemic on the crypto-market and stated that cryptocurrencies can be epidemiologically safe for transactions. This is positive feedback for accepting Bitcoin as an economically safe asset. However, there is a lack of sufficient theoretical propositions supporting Bitcoin implementation in the sustainable development of society. Therefore, this study is an attempt to develop the theoretical approach for predictive analytics for Bitcoin prices. A stable Bitcoin pricing strengthens the ESG goals and so provides support for sustainable development.

This is the first paper to perform a detailed performance evaluation to rank various models with 15 sets of evaluation criteria in Bitcoin price analysis. The major contributions of the paper are listed below.

- Twelve different predictor models were developed using six learning algorithms (five available algorithms, along with a newly proposed algorithm) in two polynomial-based neural networks.
- The predictor models are ranked in the testing phase to determine the best model that performs well in designing a Bitcoin price predictor.
- A TOPSIS-based approach is applied for ranking the models using fifteen error metrics.
- A new mutated climb monkey algorithm (MCMA) was proposed for training and testing the two neural networks under study.

Section 2 provides a detailed survey of the literature used in this study, such as Bitcoin pricing, neural networks, MA, and TOPSIS. Section 3 introduces the architecture of this study, along with explanations of related techniques and the TOPSIS methodology. Section 4 explains the proposed work and analyzes the experimental setup of the model. In Section 5, the experimental results are thoroughly analyzed. In Section 6, the study is extensively discussed and Section 7 describes the best model in this study and the future scope of this work.

2. Materials and Methods

2.1. Literature Review

Bitcoin pricing is an active research area, as very little work has been done as the Bitcoin market is not yet mature. Many studies have been conducted to understand the pricing mechanism of Bitcoin. Jang and Lee [7] proposed the use of a Bayesian neural network simulated using blockchain information as a reliable predictor model for Bitcoin prices. This study indicates that the model provides a more accurate prediction of the direction of Bitcoin prices. It also provides promising results for the recent data. Huang et al. [8] efficiently predicted the daily return ranges of Bitcoin prices. They used a classification-tree-based model to generate accurate price intervals. In addition to closing prices, many other features of Bitcoin prices have been used in different studies. Mallqui and Fernandes [9] used various machine-learning techniques for different types of price prediction. Artificial neural networks (ANNs), support virtual machines (SVMs), and recurrent neural networks (RNN) are used to predict the maximum, minimum, and closing prices. They used an SVM and an ensemble approach to analyze the direction prediction for the classification of Bitcoin directions. The highly volatile nature of cryptocurrencies, such as Bitcoin, is revealed by a study that compares the performance of Bitcoin with traditional currency. Peng et al. [10] proposed an SVM-based model to present this comparison and prove the volatility of Bitcoin prices. Aggarwal et al. [11] explained Bitcoin prediction using an SVM and complete empirical ensemble mode decomposition. Recently, when deep learning has dominated the research scenario, Liu et al. [12] have justified that stacked denoising autoencoders predict the Bitcoin price better than SVM and back-propagation neural networks.

Bitcoin price analysis is in its initial phase. Hence, very few networks have been studied using Bitcoin datasets. The FLANN model is best suited to maintain a simple architecture. Chebysev, Hermite, or Legendre polynomials can be used to generate the FLANN. Many studies have been conducted in the field of networks using other financial data. Troumbis et al. [13] effectively demonstrated a strong approximation of the Chebysev polynomial neural network models for non-linear and highly complex environmental data. They compared this model with the Hermite polynomial-based FLANN and radial basis function neural network (RBFNN). Mohanty and Dash [14] recently demonstrated the effectiveness of a Chebysev polynomial neural network trained with a flower pollination algorithm in forecasting net asset values. This was compared with basic particle swarm optimization (PSO) and differential evolution (DE). The use of Chebyshev networks in the medical field is promising. Zhou et al. [15] used Chebyshev networks to efficiently link acute hypertension with intensive care unit (ICU) admission and compared them with recurrent neural networks and convolutional neural networks. Dash [16,17] demonstrated the use of Chebyshev network with an improved frog-leaping algorithm and DE to predict stock prices accurately and also perform better statistically. The performance capability of Chebyshev FLANN was also described by Nanda et al. [18] for the accurate prediction of complex time-series models, such as rainfall in India. They have shown that FLANN gives less absolute average percentage error for the measured rainfall data in comparison to Auto-Regressive Integrated Moving Average (ARIMA) model.

The prediction capabilities of Legendre polynomial-based neural networks have also been explored by many researchers. Liu and Wang [19] used an improved Legendre network with a random time-strength function to efficiently predict the movement of the stock market index. Dash [20] uses a recurrent Legendre network for efficient foreign exchange forecasting. The orthogonal behavior of Legendre polynomials contributes to accurate results. Afifi and Zanaty [21] have also shown that Legendre kernels result in more efficient SVMs than other kernels, such as GA. Dash and Dash [22] proposed a Legendre network-based modified differential harmony search (MDHS-LPNN) model as an efficient and robust hybrid predictor for foreign exchange forecasting.

In 2008, Zhao and Tang [23] proposed a population-based nature-inspired algorithm which modeled the mountain climbing of monkeys to solve global numerical optimization problems. It is called the Monkey algorithm (MA). It consists of three processes: climb, watch-jump, and somersault. This results in a faster convergence rate for non-linear and high-dimensional problems. It uses very few parameters and, hence, is easy to implement. Soon after many researchers explored the effectiveness of MA in diverse areas, many modifications have been proposed. Zhou et al. [24] proposed an improved version of MA to solve the 0–1 knapsack problem. They introduced a greedy strategy, followed by a cooperation process immediately after Watch-Jump. The feasibility and convergence rates are improved. Marichelvam et al. [25] proposed a hybrid MA with two sub-populations working in different processes to model the scheduling of a flow shop. This algorithm is flexible, reliable, and natural. Abiyev and Tunay [26] proposed a modified monkey algorithm (MA2) to study certain benchmarking functions for solving optimization problems. This algorithm adds two additional processes: one-component perturbation and all-component perturbation. It significantly decreased the training time but provided better learning with faster convergence than MA. Zheng [27] proposed an improved monkey algorithm with dynamic adaptation (DAMA). This algorithm makes use of the chaotic search instead of a random initialization of parameters. This algorithm allowed the dynamic updating of parameters based on their runtime performance, and hence was adaptive in its operation. Chen et al. [28] proposed the use of an artificial bee colony algorithm (ABCMA) search operator with the climb process to reduce the number of climbs, and hence, become faster. This algorithm showed improved performance compared with the basic MA. Sun et al. [29]

explored the hierarchical arrangement of the Climb, Watch, and Somersault processes and allowed self-organizing of the hierarchical structure as a result of adapting to the working environment. It uses a time-varying parameter to initiate the process of self-organization. The proposed algorithm is highly effective and robust.

Zulqamain et al. [30] demonstrated the step-by-step use of the technique for order of preference by similarity to an ideal solution (TOPSIS) approach to select the best automated car based on multiple criteria. Cocis et al. [31] compared the TOPSIS ranking of 22 airline industries based on 8 financial indicators with Fortune Ranking and concluded that TOPSIS provided accurate rankings for 3 airlines. Batrancea et al. [32] explained that the increase in a number of financial indicators generated better ranks for airline industries under study using an MCDM ranking. Sabaghi et al. [33] proposed a modified TOPSIS method that can be used to add or remove alternatives during the study by compromising the reliability to some extent. Samal and Dash [34] proposed a TOPSIS-based extreme learning machine (TOPSIS_ELM) framework to predict stock price movements. This network is quite robust in nature and is more reliable, as it uses evaluations based on multiple criteria. Dash et al. [35] demonstrated the use of TOPSIS in ranking 13 classifiers and seven evaluation measures to select the best classifier ensemble for forecasting stock-price movements.

2.2. Theoretical Background

In the literature, very few applications of computational techniques have appeared for different aspects of crypto-market analysis, such as forecasting Bitcoin price, its movement, its volatility, and so on. Among the various techniques, artificial neural networks (ANNs) are more widely used. Although various types of ANN have provided good results in regression analysis, most of them suffer from diminished generalization, complex network structures, continuous parameter tuning, and the need for efficient training algorithms. Determining the architecture of the network, learning algorithms to be used, and selecting proper evaluation strategies have always been challenging for researchers.

Many studies have been conducted to efficiently predict Bitcoin prices using machine-learning and deep-learning techniques. However, to address the high volatility and large solution space, new learning algorithms need to be explored.

Neural networks are more flexible for teaching large datasets and provide more accurate approximations. Functional link neural networks (FLANNs) are simple to implement because of their clear architecture. Because the Bitcoin market requires a faster and more accurate evaluation, the features of FLANN can help in designing an efficient model. Machine learning algorithms have been implemented in FLANNs for stock price prediction, gold price prediction, and many more. However, the efficacy of FLANN in Bitcoin prediction is yet to be analyzed. This study utilizes the simple architecture of polynomial-based FLANNs to predict Bitcoin prices.

The search domain of Bitcoin prices is very large, owing to its high volatility. Hence, a learning algorithm which can predict a larger search domain is essential. Several nature-inspired algorithms have been successfully used to model real-world problems. Such algorithms can also help in forecasting Bitcoin prices. In addition, the need to use a nature-inspired algorithm with a broader search domain is the motivation to select the monkey algorithm (MA). Hence, each of the six learning techniques used in the two different neural networks resulted in 12 different models for Bitcoin price forecasting. Therefore, developing an efficient predictor by analyzing the effect of 12 different models that can produce better results with respect to multiple performance criteria can be considered a Multi-Criteria Decision-Making (MCDM) problem. In this study, the 12 models were evaluated using 15 performance metrics and the multi-criteria decision-making algorithm TOPSIS. The 15 error measures included the root mean square error (RMSE), mean absolute error (MAE), mean square error (MSE), R-square error (RSQR), Theil's U error (TU), mean percentage error (MPE), mean absolute error percentage (MAEP), symmetric mean absolute percentage error (SMAPE), mean absolute square error (MASE), sum square error (SSE), root squared sum error (RSSE), mean relative absolute error (MRAE), mean signed deviation (MSD), root relative square error (RRSE), and average relative variance (ARV). Therefore, this paper presents a 12 × 15 MCDM problem with 12 models and 15 criteria. The models were evaluated by ranking them during the testing phase. The study is carried out by using historical closing prices of Bitcoin in three currencies: US Dollar (USD), European Euro (EUR), and Japanese YEN (JPY). The new mutated climb monkey algorithm (MCMA) provides impressive results compared with other models.

3. Related Work

3.1. PBNN

A polynomial-based neural network (PBNN) is a breakthrough in neural network implementation. Unlike having multiple hidden layers, the PBNN provides a devoted input-widening unit which converts the input data into a wide spectrum of values using a defined polynomial representation. By doing this, the non-linear relationship between the input and output data is mapped. This study used two PBNN: Chebyshev PBNN (CPBNN) and Legendre PBNN (LPBNN). Figure 1 depicts the architecture of the PBNN.

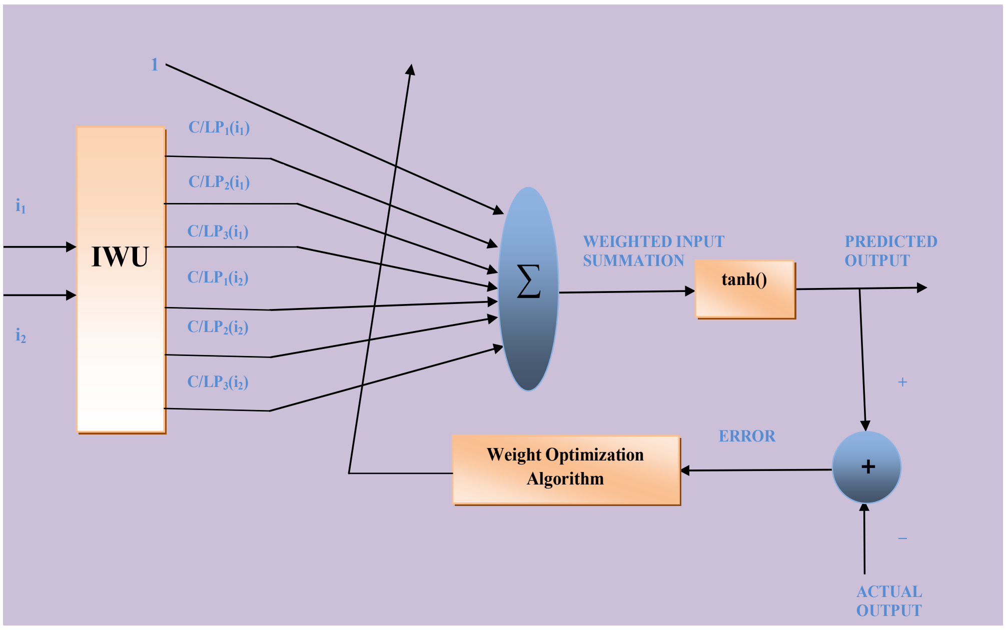

Figure 1. Architecture of Polynomial Based Neural Network.

In a PBNN, the input data are expanded using an input widening unit (IWU) based on the underlying Chebyshev or Legendre polynomials. These expanded inputs are then passed through a summation and activation unit to generate predicted outputs. In this study, tanh() was used as the activation function. The predicted output was compared to the actual output to generate an error measure. This error had been used by different weight optimization algorithms for efficient weight updates. The optimized weights generated by each model after training can be further utilized for testing.

The CPBNN uses Chebyshev polynomials for input expansion. These polynomials are represented recursively as in Equation (1).

$$\mathrm{CP}_{j+1}(i) = 2i\mathrm{CP}_j(i) - \mathrm{CP}_{j-1}(i) \quad (1)$$

where $\mathrm{CP}_j(i)$ represents the Chebyshev polynomial of order j and input $-1 < i < 1$. The initial values were $\mathrm{CP}_0(i) = 1$ and $\mathrm{CP}_1(i) = i$.

The LPBNN uses Legendre polynomials for input expansion. These polynomials are represented recursively as in Equation (2).

$$LP_{j+1}(i) = \frac{1}{j+1}\left[(2j+1)iLP_j(i) - jLP_{j-1}(i)\right] \quad (2)$$

where $LP_j(i)$ represents the Legendre polynomial of order j and input $-1 < i < 1$. The initial values were $LP_0(i) = 1$ and $LP_1(i) = i$.

3.2. Mutated Climb Monkey Algorithm

The MCMA algorithm proposed in this study was inspired by a meta-heuristic optimization process which imitates the behavior of monkeys. The basic MA consists of three steps: the Climb, Watch-Jump, and Somersault steps.

3.2.1. Climb Process

In this process, the climbing behavior of monkeys was modeled to reach the mountain top. The maximum number of climbs represents the strength of the monkey to climb in one go. This is also affected by the length of each step when the monkey climbs. A significant amount of time is spent on the climbing process if the step length is very small and the number of climbs is greater.

The climb process is used to modify the positions of monkeys from the starting positions to the updated positions by adding an improvement to the objective function. Initially, let each monkey has an initial position x_i, for i = 1,2, . . . ,M, for M monkeys. The climbing process of monkey i is as follows:

1. Randomly generate r_i for each monkey position such that $r_i = s$ or $r_i = -s$ with equal probability, where s is the step length.
2. Calculate the pseudo-gradient of the fitness function f(x) using Equation (3).

$$gradf_i(x_i) = \frac{f(x_i + r_i) - f(x_i - r_i)}{2r_i} \quad (3)$$

3. Calculate a new monkey position y using

$$y_i = x_i + s.gradf_i(x_i) \quad (4)$$

4. Update the monkey position to y_i (a climb) if its fitness value of a new position is better than that of the previous position.
5. Repeat Steps 1 to 4 until the maximum number of climbs is reached.

3.2.2. Watch-Jump Process

After reaching the local mountain top, each monkey uses its eyesight to look for more high mountains (watch), and then jumps to the best one (jump). Again, the monkey starts climbing to the top of the jumped mountain. This is repeated until the number of jumps is available. The steps of the watch-jump are as follows:

1. Randomly generate a new location y_i from the interval (x_i + e, x_i − e) where e represents the eyesight of the monkey.
2. If the fitness of the monkey at y_i is better than that of current position then update the monkey position to y_i (A Jump).
3. Repeat steps 1 and 2 till maximum number of jump is reached.

3.2.3. Somersault Process

After obtaining the optimal solution in a search domain, the monkey searches for a new domain that is available to explore the global best position. To do a somersault, the barycenter of the current positions of all monkeys was identified as a pivot. After this, the rest of the monkeys somersault in the pivot direction. The somersault steps are as follows:

1. Generate a random number z from the somersault interval (a, b).
2. Considering M as the monkey population find a pivot location P_i given by Equation (5).

$$P_i = \frac{\left(\sum_{i=1}^{M} x_i\right)}{M} \quad (5)$$

3. Find a new position y_i using Equation (6).

$$y_i = x_i + z \times (P_i - x_i) \quad (6)$$

4. If the fitness of the monkey at y_i is better than that of the previous position then update the monkey position to y_i.
5. Repeat steps 1–4 until the maximum number of somersaults is reached.

The working of the MA greatly depends on the climbing process. The MA algorithm requires more time for climbing if the step-length is smaller, whereas a larger step-length can miss some optimal positions. Many researchers have used updated climbs to generate good MA results. In the DE algorithm, the mutation operation generates a mutated position which can provide optimized results. Hence, if we apply a mutation on the climb feature, the mutated monkey position can result in a better solution and can converge faster, instead of moving each step-length. The MCMA algorithm proposed here improves the performance of the basic MA algorithm by adding a mutated version of climb such that the best climb is used in the optimization process. The mutated climb position mx_i is calculated using

$$mx_i = x_{r3} + m_{scale} \times (r_1 - r_2) \quad (7)$$

Here, r1, r2, and r3 are three random numbers between 1 and M and m_{scale} represents the mutation scale. In the proposed algorithm, a mutated climb position is generated before each climb. If the fitness value of the mutated climb is better than that of the previous climb, the mutated climb position mx_i is attained by the monkey i. This results in faster convergence and faster climbing of the monkeys.

The stepwise representation of MCMA is presented in Algorithm 1.

3.3. TOPSIS

In this study, models for Bitcoin price prediction were evaluated using 15 different evaluation criteria. Hence, it can be described as a multi-criteria decision-making (MCDM) problem with 12 models and 15 criteria. This problem is solved using the TOPSIS approach.

In the TOPSIS approach, the best model is selected based on the minimum distance from the positive-ideal model and the maximum distance from the negative-ideal model. Positive model values are the best values for each criterion, and the negative model values represent the worst values for each criterion. The steps of ranking the 12 predictors using 15 different criteria in the TOPSIS approach can be summarized as follows:

1. Generate a standardized decision table (SDT) for 12 rows representing each model and 15 columns representing each criterion.
2. Assign weights to each criterion and generate a weighted SDT.
3. Create a positive ideal model with the best values for each criterion and a negative ideal model with the worst values.
4. Calculate the distance of each model from the positive and negative ideals using 15-dimensional Euclidean distance.
5. Find the relative closeness of each model.
6. Relative closeness is arranged in decreasing order to generate the ranking of the models.

Algorithm 1: MCMA.

Input: Initial monkey positions (weights), say, x_i, max-iterations, expanded training input, and output.
Output: Updated weight values (monkey positions)
Begin
1. Initialize the parameter step-length s, eyesight e, somersault interval (a,b), max-climb N_c, max-jump N_j, and max-somersault N_s.
2. Calculate the fitness value of the initial population and determine the current best solution, for example, *g_best*.
3. **while** (max-iterations not reached) **do**
4. **while** (N_c not reached) **do**
5. **for** (each monkey position x_i) **do**
6. **Apply mutation to generate a mutated position** mx_i by using (7).
7. **if** (fitness of mx_i is better than x_i) **then**
8. Update x_i with mx_i.
9. end if
10. Calculate the pseudo-gradient by using (3).
11. Calculate a new position y_i by using (4).
12. if (fitness of y_i is better than x_i) **then**
13. Update x_i with y_i.
14. end if
15. end for
16. end while
17. Find the current best solution *c_best*.
18. if (*c_best<g_best*) **then**
19. g_best = c_best;
20. end if
21. while (N_j not reached) **do**
22. for (each monkey position x_i) **do**
23. Randomly Generate a new location y_i from the interval (x_i + e, x_i − e)
24. if (fitness of y_i is better than x_i) **then**
25. Update x_i with y_i.
26. end if
27. end for
28. end while
29. Find the current best solution *c_best*.
30. if (*c_best<g_best*) **then**
31. g_best = c_best;
32. goto step 4 with climb = 1(counting the climbs again)
33. end if
34. while (N_s not reached) **do**
35. Generate a random number z from the interval (a, b).
36. Find a pivot location P_i by using (5).
37. Find a new position y_i by using (6).
38. if (fitness of y_i is better than x_i) **then**
39. Update x_i with y_i.
40. end if
41. end while
42. Find the current best solution *c_best*.
43. if (*c_best<g_best*) **then**
44. g_best = c_best;
45. end if
46. end while
47. The monkey position *g_best* giving the best fitness value is the optimized result.
End

The TOPSIS approach used in this study determines a better predictor using the following performance criteria:

Assume that P(i) and A(i) represent the ith predicted price and ith actual price, respectively.

- Root Mean Square Error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(P(i) - A(i))^2}{n}} \tag{8}$$

- Mean Absolute Error (MAE)

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}|P(i) - A(i)|}{n} \tag{9}$$

- Mean Square Error (MSE)

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n}(P(i) - A(i))^2}{n} \tag{10}$$

- R-Square Error (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(A(i) - P(i))^2}{\sum_{i=1}^{n}\left(A(i) - \overline{A}\right)^2} \tag{11}$$

- Theil'sU Error (TU)

$$\mathrm{TU} = \frac{\sqrt{\frac{1}{n} \times \sum_{i=1}^{n}(P(i) - A(i))^2}}{\sqrt{\frac{1}{n} \times \sum_{i=1}^{n} A(i)^2} + \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} P(i)^2}} \tag{12}$$

- Mean Percentage Error (MPE)

$$\mathrm{MPE} = \frac{100}{n} \times \sum_{i=1}^{n} \frac{A(i) - P(i)}{A(i)} \tag{13}$$

- Mean Absolute Error Percentage (MAEP)

$$\mathrm{MAEP} = \frac{\sum_{i=1}^{n}|P(i) - A(i)|}{n} \times 100 \tag{14}$$

- Symmetric Mean Absolute Percentage Error (SMAPE)

$$\mathrm{SMAPE} = \frac{100}{n} \times \sum_{i=1}^{n} \frac{|P(i) - A(i)|}{\frac{|A(i)| + |P(i)|}{2}} \tag{15}$$

- Mean Absolute Scaled Error (MASE)

$$\mathrm{MASE} = \frac{\frac{1}{n} \times \sum_{i=1}^{n}|A(i) - P(i)|}{\frac{1}{n-1} \times \sum_{i=2}^{n}|A(i) - A(i-1)|} \tag{16}$$

- Sum Squares Error (SSE)

$$\mathrm{SSE} = \sum_{i=1}^{n}(A(i) - P(i))^2 \tag{17}$$

- Root Squared Sum Error (RSSE)

$$\mathrm{RSSE} = \sqrt{\sum_{i=1}^{n}(A(i) - P(i))^2} \quad (18)$$

- Mean Relative Absolute Error (MRAE)

$$\mathrm{MRAE} = \frac{1}{n} \times \sum_{i=2}^{n}\left|\frac{A(i) - P(i)}{A(i) - A(i-1)}\right| \quad (19)$$

- Mean Signed Deviation (MSD)

$$\mathrm{MSD} = \frac{1}{n} \times \sum_{i=1}^{n}(A(i) - P(i)) \quad (20)$$

- Root Relative Square Error (RRSE)

$$\mathrm{RRSE} = \sqrt{\frac{\sum_{i=1}^{n}(P(i) - A(i))^2}{\sum_{i=1}^{n}\left(P(i) - \overline{A}\right)^2}} \quad (21)$$

- Average Relative Variance (ARV)

$$\mathrm{ARV} = \frac{\sum_{i=1}^{n}(P(i) - A(i))^2}{\sum_{i=1}^{n}\left(P(i) - \overline{A}\right)} \quad (22)$$

The listed error measures were evaluated to identify the best models. The R^2 measure is a positively oriented error measure, as higher values are better. The remaining 14 error measures were negatively oriented, as lower values were preferred.

4. Proposed Work

Figure 2 depicts the architecture of the proposed model for Bitcoin pricing.

The work starts with the collection of Bitcoin closing price data and uses it as a Bitcoin dataset. This dataset is then passed through the data pre-processing steps which include data normalization using the max-min normalization technique, followed by the creation of input data and output data using the windowing technique. Then, the data were further divided into training and testing data in a 2:1 ratio. The training data acts as an input for the next step. Here, each of the CPBNN and LPBNN networks is trained using six learning algorithms: PSO, DE, MA, MA2, ABCMA, and the newly proposed MCMA. The optimized weight generated after training the network is used to predict the Bitcoin price in the next step. In the next step, the expanded testing data were used to test the predictive capabilities of 12 models, that is, CPSO, CDE, CMA, CMA2, CABCMA, CMCMA, LPSO, LDE, LMA, LMA2, LABCMA, and LMCMA models. Then, in the next step, the 15 performance measures, that is, RMSE, MAE, MSE, R^2, TU, MPE, MAEP, SMAPE, MASE, SSE, RSSE, MRAE, MSD, RRSE, and ARV were calculated. To obtain the best model for prediction and analyze the prediction features of the proposed learning technique a 15-criterion-based MCDM approach called TOPSIS was used. This approach ranks all the 12 models under consideration. The new mutated climb monkey algorithm (MCMA) was proposed as the best Bitcoin predictor after analysis. All experiments were performed in a system with an Intel® Core(TM) i3-4005U CPU @ 1.70 GHz processor, 4 GB RAM, and Windows 8.1 pro 64-bit operating system. The implementation of all models and their ranking were performed using MATLAB R2014a.

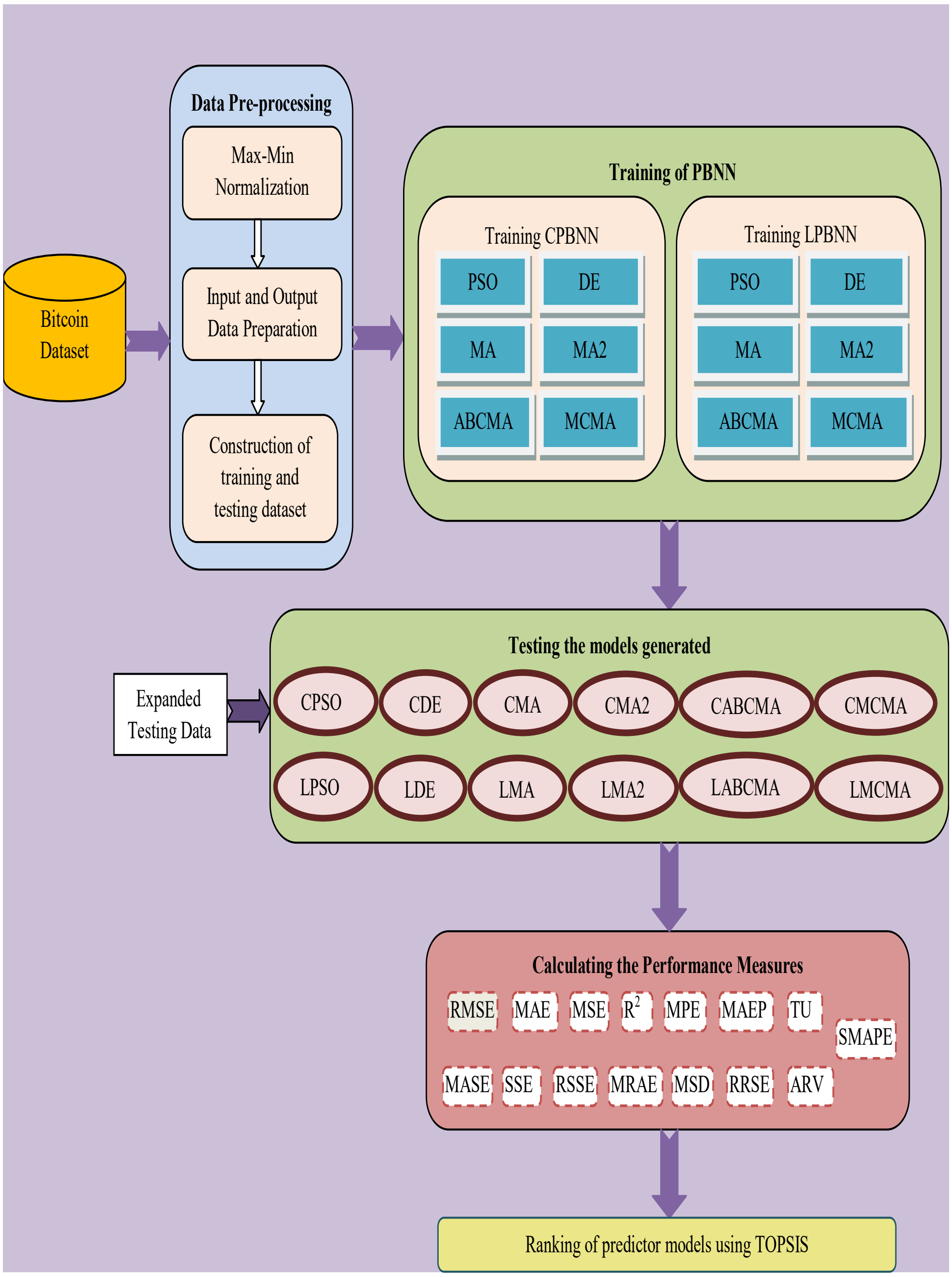

Figure 2. Proposed Polynomial Based Neural Network Bitcoin price predictor trained using a new Mutated Climb Monkey Algorithm.

4.1. Experimental Setup

4.1.1. Datasets

The datasets used in this study are accessed from http://www.investing.com/crypto/bitcoin/historical-data, accessed on 15 June 2021. The datasets indicate the daily closing price of Bitcoin in US dollars (BTC/USD), European euro (BTC/EUR), and Japanese yen (BTC/JPY) for the time range from 14 May 2017 to 14 June 2021. As the research idea was formulated in May 2021, the most recent Bitcoin prices then were considered for the study. It was observed that Bitcoin prices started to have a steady growth from April 2017 and followed a constant upward trend till March 2021 and were then gradually falling till the date of data collection. Therefore, this time frame was selected to get the most volatile prices. This timeframe can also provide the effect of the COVID-19 pandemic on the crypto-market. Each dataset was divided into training and testing data. The details of each dataset are listed in Table 1.

Table 1. Dataset Descriptions.

Data Sets	Total Samples	Training Samples	Testing Samples
BTC/USD	1493	992	496
BTC/EUR	1493	992	496
BTC/JPY	1493	992	496

4.1.2. Parameter Setup

The values assigned to each parameter for the various learning techniques used in this study are listed in Table 2.

Table 2. Parameter Descriptions.

Learning Methods	Parameters	Values
PSO	Inertia	0.4
	C1	0.9
	C2	0.9
DE	Mutation Factor	0.4
	Crossover coefficient	0.9
MA/MA2/ABCMA	Maximum Climb	10
	Maximum Jump	5
	Maximum Somersault	2
	Eyesight	0.4
	Step-length	0.1
	Somersault-Interval	0.4–0.6
MCMA	Maximum Climb	10
	Maximum Jump	5
	Maximum Somersault	2
	Eyesight	0.4
	Step-length	0.1
	Somersault-Interval	0.4–0.6
	Mutation Factor	0.4

4.1.3. Procedural Analysis

To study the forecasting ability of the proposed algorithm, datasets of Bitcoin prices in dollar, euro, and yen were collected and processed. The datasets used are listed in Table 1. To obtain a uniform search space, the data were normalized using the maximum and minimum values in each dataset. The uniform data were then divided into training and testing data at a ratio of 2:1. A neural network with five input neurons and one output neuron was designed to train the datasets. Such a network uses the windowing technique to predict the sixth-day Bitcoin price by using the last five days' price. Similarly, a sliding window of size 5 slides through the entire dataset with prediction horizon 1 to generate the input and output datasets. Initially, the CPBNN and LPBNN trained each of the three datasets using PSO, DE, MA, MA2, ABCMA, and MCMA algorithms. This resulted in 12 trained models on the three different training datasets. The RMSE value was used as the fitness function for each model. Each model uses a population size of 20 and is executed hundred times iteratively to predict the one-day-ahead Bitcoin price on the three testing datasets. The performance of each model was evaluated using 15 performance measures: RMSE, MAE, MSE, RSQR, TU, MPE, MAEP, SMAPE, MASE, SSE, RSSE, MRAE, MSD, RRSE, and ARV. The minimum value of each performance measure was used as the final error measure value after executing the models ten times each.

The third phase of the proposed procedure uses the TOPSIS algorithm to rank the 12 models using the 15 error measure values as the criteria for evaluation. This process acts as a multi-criteria ranking method for the 12 models: CPSO, CDE, CMA, CMA2,

CABCMA, CMCMA, LPSO, LDE, LMA, LMA2, LABCMA, and LMCMA. In the literature, there are many forecasting models which are evaluated based on certain error measures. However, models may result in different efficiency for these measures. In this study, there are 12 models for comparison but the value of the 15 error measures used for evaluation vary from model to model as shown in Tables 3–5. Therefore, to choose the best model a multi-criteria decision-making algorithm can be useful.

Many applications of the MCDM technique have been studied in the automobile and airline industries [30,31]. It has been observed that TOPSIS is a highly reliable and accurate MCDM method as compared to other MCDM methods [31,33]. Additionally, the TOPSIS ranking considers the distance from both the positive ideal solution and negative ideal solution for selecting the fittest alternative and hence gives a better decision. Though TOPSIS has to be redone if more alternatives are included or excluded, this study uses a fixed 12 set of models as alternatives. TOPSIS has proved to compare regression and classification models more accurately and robustly [33–35]. Therefore, to select the most accurate and reliable Bitcoin pricing model out of the 12 different models under study, a 15-criterion-based TOPSIS analysis was used. The models were ranked using TOPSIS for all three datasets.

Table 3. Performance Metrics for BTC/USD dataset.

Models	RMSE	MSE	MAE	TU	R^2	MPE	MAPE	SMAPE	MASE	SSE	RSSE	MRAE	MSD	ARV	RRSE
CPSO	0.0557	0.0031	0.0328	0.1227	0.6517	−45.81	9.4172	9.4828	2.8193	1.5374	1.2399	12.778	−0.028	−0.154	0.1757
CDE	0.0405	0.0016	0.0246	0.0893	0.1054	−11.86	6.3815	6.0784	2.1163	0.8132	0.9018	14.615	−0.146	−0.522	0.1396
CMA	0.0499	0.0025	0.0272	0.11	0.7809	−12.66	7.0582	6.6907	2.342	1.2358	1.1117	25.392	0.0145	0.1576	0.1617
CMA2	0.0624	0.0039	0.039	0.1376	0.8833	−13.27	10.046	9.2536	3.353	1.9315	1.3898	12.91	0.0339	0.1025	0.193
CABCMA	0.0508	0.0026	0.0295	0.1119	0.8885	−15.17	7.1978	6.7071	2.54	1.2792	1.131	12.91	0.0245	**0.0845**	0.1626
CMCMA	0.0359	0.0013	0.021	0.0792	**0.9025**	−13.95	**5.2051**	**5.265**	1.8049	0.6398	0.7998	**7.3807**	−0.014	−0.091	0.1317
LPSO	0.0451	0.002	0.0257	0.0995	0.7096	−28.58	7.2755	6.983	2.2127	1.0099	1.0049	9.945	**−0.0039**	−0.731	0.1507
LDE	0.0367	0.0013	0.0215	0.0809	0.4597	−12.32	5.4609	5.3083	1.8512	0.6684	0.8175	9.6553	−0.126	−1.832	0.1327
LMA	0.0718	0.0052	0.0438	0.1582	0.8498	**−11.149**	10.596	10.098	3.7643	2.5557	1.5986	14.376	0.0341	0.1511	0.2117
LMA2	0.0344	0.0012	0.0221	0.0758	0.8097	−14.18	6.2822	6.0527	1.9033	0.586	0.7655	39.278	0.0081	0.1464	0.1197
LABCMA	0.0749	0.0056	0.0445	0.1651	0.8365	−15.84	10.318	9.3598	3.8257	2.7811	1.6677	54.048	0.0376	0.1307	0.2231
LMCMA	**0.032**	**0.001**	**0.0201**	**0.0706**	0.5256	−12.34	6.7771	6.5198	**1.7257**	**0.5085**	**0.7131**	7.4666	-0.094	−0.429	**0.106**

Table 4. Performance Metrics for BTC/EUR dataset.

Models	RMSE	MSE	MAE	TU	R^2	MPE	MAPE	SMAPE	MASE	SSE	RSSE	MRAE	MSD	ARV	RRSE
CPSO	0.0378	0.0014	0.0237	0.0838	−0.461	−22.57	6.9309	6.8292	2.0818	0.7089	0.842	**11.224**	−0.2	−0.605	0.1344
CDE	0.0478	0.0023	0.0259	0.106	0.8704	−7.929	**5.2931**	5.3937	2.2681	1.1347	1.0652	34.467	−0.023	−0.123	0.1709
CMA	0.0398	0.0016	0.0242	0.0881	0.8807	−16.16	6.8146	6.73	2.126	0.7839	0.8854	14.084	−0.01	−0.163	0.1395
CMA2	0.0409	0.0017	0.0246	0.0906	0.8888	−16.25	6.7622	6.3639	2.1573	0.8283	0.9101	12.368	0.0206	0.0811	0.1361
CABCMA	0.0332	**0.0011**	**0.0198**	0.0736	0.8617	−14.73	5.7277	5.546	**1.7412**	0.5475	0.74	12.368	0.0048	**0.0636**	0.1133
CMCMA	**0.0331**	**0.0011**	0.0233	**0.0733**	0.8648	−17.584	7.5087	7.4859	2.0443	**0.5432**	**0.737**	11.374	−0.0178	−0.1443	**0.12**
LPSO	0.0648	0.0042	0.0412	0.1436	0.6894	−34.16	10.68	9.7407	3.6146	2.0816	1.4428	21.439	−0.079	−0.235	0.197
LDE	0.036	0.0013	0.0225	0.0797	**0.9295**	−12.65	6.8651	6.9805	1.9711	0.6414	0.8009	11.291	−0.022	−0.155	0.1182
LMA	0.0568	0.0032	0.0334	0.1258	0.8195	−15.78	8.5525	7.9187	2.9327	1.5996	1.2647	18.921	0.0314	0.1027	0.1805
LMA2	0.0586	0.0034	0.0344	0.1299	0.8335	−11.49	8.5829	8.2202	3.0164	1.7053	1.3059	54.641	0.0257	0.1336	0.1815
LABCMA	0.0357	0.0013	0.0213	0.0792	0.8062	−13.97	6.2202	6.0306	1.8668	0.6339	0.7961	81.493	0.0038	0.1292	0.123
LMCMA	0.0345	0.0012	0.0206	0.0764	0.911	**−6.8701**	5.3417	**5.1317**	1.803	0.5902	0.7683	29.463	**0.0045**	0.1056	0.1189

Table 5. Performance Metrics for BTC/JPY dataset.

Models	RMSE	MSE	MAE	TU	R^2	MPE	MAPE	SMAPE	MASE	SSE	RSSE	MRAE	MSD	ARV	RRSE
CPSO	0.04	0.0016	0.0229	0.0893	0.7435	−20.41	7.1597	6.7312	2.0012	0.7942	0.8912	7.9884	−0.019	−0.53	0.132
CDE	0.0333	0.0011	0.0206	0.0744	0.7234	−11.18	6.0212	5.859	1.8033	0.5513	0.7425	6.3123	−0.071	−0.33	0.1161
CMA	0.0316	**0.001**	0.0183	0.0705	0.8729	−12.64	4.871	4.8051	1.5977	0.4956	0.704	5.5411	−0.009	−0.106	0.1144
CMA2	0.0377	0.0014	0.0214	0.0841	0.8071	−9.796	5.9668	5.7162	1.8714	0.7045	0.8394	18.197	0.0104	0.1035	0.1245
CABCMA	0.0421	0.0018	0.0241	0.0939	**0.9108**	−8.599	6.5601	6.2543	2.1139	0.8772	0.9366	5.5411	0.012	**0.0871**	0.1361
CMCMA	0.0307	0.0009	**0.0176**	0.0685	0.6492	−9.8339	**4.7559**	**4.5433**	**1.5386**	0.4676	0.6838	**4.178**	−0.0844	−0.3528	0.1039
LPSO	0.0322	**0.001**	0.0188	0.0718	0.7548	−17.44	5.0267	4.9054	1.6428	0.5139	0.7169	5.9969	−0.055	−0.827	0.1127
LDE	0.0336	0.0011	0.019	0.0751	0.9013	**−6.1986**	5.1167	4.9142	1.6598	0.5611	0.749	14.101	**0.0022**	0.1091	0.1115
LMA	0.0518	0.0027	0.0298	0.1157	0.8547	−11.645	8.1008	7.8012	2.6073	1.3334	1.1547	8.9738	0.0199	0.1137	0.166
LMA2	0.045	0.002	0.025	0.1004	0.826	−13.79	6.759	6.5186	2.1881	1.0034	1.0017	29.062	0.0135	0.1218	0.1455
LABCMA	0.0557	0.0031	0.0345	0.1244	0.8505	−10.12	8.3642	7.8741	3.0162	1.2414	1.5412	42.573	0.0299	0.0909	0.1773
LMCMA	**0.0274**	0.0008	0.0179	**0.0612**	0.7484	−28.57	5.972	6.0598	1.5634	**0.373**	**0.6108**	7.1639	−0.004	−0.201	**0.0947**

5. Result Analysis

As this study uses a supervised learning technique, all 12 models, that is, CPSO, CDE, CMA, CMA2, CABCMA, CMCMA, LPSO, LDE, LMA, LMA2, LABCMA, and LMCMA, were trained using the training samples available for each of the three datasets under consideration. After training the models, the optimized weights were used to generate the predicted Bitcoin prices for the test samples. This prediction was repeated 10 times for each model. The minimum error measure values for the 15 performance metrics after 10 runs of each of the 12 models in the testing samples of the BTC/USD, BTC/EUR, and BTC/JPY datasets used in this study are listed in Table 3, Table 4, and Table 5, respectively.

From Table 3, representing the BTC/USD dataset, it can be observed that the LMCMA model provided the minimum values for RMSE, MSE, MAE, TU, MASE, SSE, RSSE, and RRSE. However, the performance of the CMCMA model on this dataset was the best in terms of R^2, MAPE, SMAPE, and MRAE. However, it can be clearly seen that the LMA model gives the best value for the MPE, the LPSO model minimizes the MSD metric, and the CABCMA model gives the best ARV value.

From Table 4, which represents the BTC/EUR dataset, it can be seen that the CMCMA model resulted in the minimum values for RMSE, MSE, TU, SSE, RSSE, and RRSE. For the same dataset, the CABCMA model showed the minimum MSE, MAE, MASE, and ARV. In addition, the LMCMA model provides the minimum values for MPE, SMAPE, and MSD. However, the CPSO model gives the minimum MRAE value, the LDE model gives the best R^2 value, and the CDE model provides the minimum MAPE value.

From Table 5, which represents the BTC/JPY dataset, the CMCMA model provided the minimum MAE, MAPE, SMAPE, MASE, and MRAE. Here, the LMCMA model provides the minimum values for RMSE, TU, SSE, RSSE, and RRSE. In addition, CABCMA provides a minimum value for R^2 and ARV. However, the LDE model yielded the minimum MPE and MSD. In addition, both the LPSO and CMA models provided the minimum MSE value.

After the analysis of all 15 performance measures of each predictor, it was observed that none of the models performed the best. Although a few models give better results in a few criteria, they do not perform well in the rest of the criteria. Therefore, a multiple-criteria-based evaluation strategy is required to obtain the best model for Bitcoin price prediction. A TOPSIS approach was used to provide a robust analysis of the performance of each model and rank the 12 models. Hence, Tables 3–5 are used as 12×15 decision matrices for the BTC/USD, BTC/EUR, and BTC/JPY datasets, respectively. Then, it follows the TOPSIS approach to generate the ranks of each model. In addition, the evaluation of the 12 predictor models in terms of multiple criteria can provide a better explanation of the best Bitcoin predictor. Although the minimum values for each of the three datasets are promising for MCMA models, the error measurement values for the other models are also very close to the minimum value. Therefore, to identify the best predictor model in this study, the

TOPSIS approach was used. The final matrix showing the relative closeness of each model, and hence, the final ranking of models based on the 15 performance metrics using the TOPSIS approach for each of the three datasets under study are shown in Tables 6–8.

Table 6. Ranking for BTC/USD dataset.

Models	Relative Closeness	Rank
CPSO	0.6635	10
CDE	0.7062	7
CMA	0.7398	5
CMA2	0.7484	**3**
CABCMA	0.6646	9
CMCMA	0.7642	**2**
LPSO	0.7263	6
LDE	0.6848	8
LMA	0.6050	11
LMA2	0.5387	12
LABCMA	0.7436	4
LMCMA	0.8210	**1**

Table 7. Ranking for BTC/EUR dataset.

Models	Relative Closeness	Rank
CPSO	0.6912	8
CDE	0.7176	5
CMA	0.7146	6
CMA2	0.7408	4
CABCMA	0.7431	**3**
CMCMA	0.8119	**1**
LPSO	0.6827	9
LDE	0.7122	7
LMA	0.6252	11
LMA2	0.4993	12
LABCMA	0.6325	10
LMCMA	0.7938	**2**

Table 8. Ranking for BTC/JPY dataset.

Models	Relative Closeness	Rank
CPSO	0.6534	10
CDE	0.7893	5
CMA	0.7846	6
CMA2	0.6926	8
CABCMA	0.8304	**2**
CMCMA	0.8524	**1**
LPSO	0.6394	11
LDE	0.7707	7
LMA	0.6854	9
LMA2	0.7945	4
LABCMA	0.6361	12
LMCMA	0.8058	**3**

As inferred from Table 6, in the BTC/USD dataset, the LMCMA model ranked first in the analysis, followed by the CMCMA and CABCMA models. In addition, in the BTC/EUR dataset, the CMCMA model was the first, followed by the CABCMA model. In this dataset, the LMCMA model ranked third. Table 8 shows that the CMCMA model is ranked first followed by the LMCMA and CABCMA models in the BTC/JPY dataset. The model in the top three ranks for each dataset under consideration is presented in Table 9.

Table 9. Top 3 predictor models ranked using TOPSIS for three datasets.

Rank of Models	BTC/USD	BTC/EUR	BTC/JPY
1	LMCMA	CMCMA	CMCMA
2	CMCMA	CABCMA	LMCMA
3	MA2	LMCMA	CABCMA

To address the stability issues, the CMCMA model was executed for three different expansion orders ($p = 2$, $p = 3$, and $p = 4$). The minimum RMSE value for each dataset for the corresponding value of p was recorded. The values are shown in Table 10.

Table 10. Minimum RMSE for datasets using CMCMA model with different expansion order.

Models	RMSE for $p = 2$	RMSE for $p = 3$	RMSE for $p = 4$
BTC/USD	**0.0260**	0.0408	0.0468
BTC/EUR	**0.0265**	0.0360	0.0569
BTC/JPY	**0.0278**	0.0383	0.551

Table 10 indicates that the CMCMA model performs the best in an expansion order of 2. To further check the stability of the proposed model, the parameters of the MCMA model are further tuned using expansion order 2. The minimum RMSE value for each dataset with different values of N_c (maximum number of climbs) is shown in Table 11.

Table 11. Minimum RMSE for CMCMA model using different values for maximum number of climb.

MODELS	$N_C = 10$	$N_C = 20$	$N_C = 30$
BTC/USD	**0.0241**	0.0288	0.0300
BTC/EUR	**0.0273**	0.0279	0.0277
BTC/JPY	**0.0263**	0.0271	0.0281

Table 11 clearly shows that the CMCMA model gives the minimum error for $N_c = 10$ for all the datasets under study. By using the number of maximum climbs as 10, the CMCMA model was evaluated for three different values of the climb step length parameter s. The minimum RMSE values obtained for each dataset are shown in Table 12.

Table 12. Minimum RMSE for CMCMA model using different climb step length.

MODELS	s = 0.05	s = 0.1	s = 0.15
BTC/USD	0.0250	**0.0236**	0.0246
BTC/EUR	**0.0256**	**0.0256**	0.0285
BTC/JPY	0.0256	**0.0245**	0.0249

Table 12 indicates that the CMCMA model performs the best with a climb step length of 0.1. The CMCMA model was tested for stability by changing the values of various input parameters.

To gain better insight into the stable predictive capability of the top two models in each of the three datasets under consideration, the predicted prices during both the training and testing of the models are plotted along with the actual price. Figures 3 and 4 show the training, testing, and actual prices for the BTC/USD dataset executed using the LMCMA and CMCMA models, respectively. It was observed that the dynamic changes in Bitcoin prices were successfully trained using both models. However, during testing, the LMCMA model provided a better approximation of the Bitcoin price than the CMCMA model for this dataset.

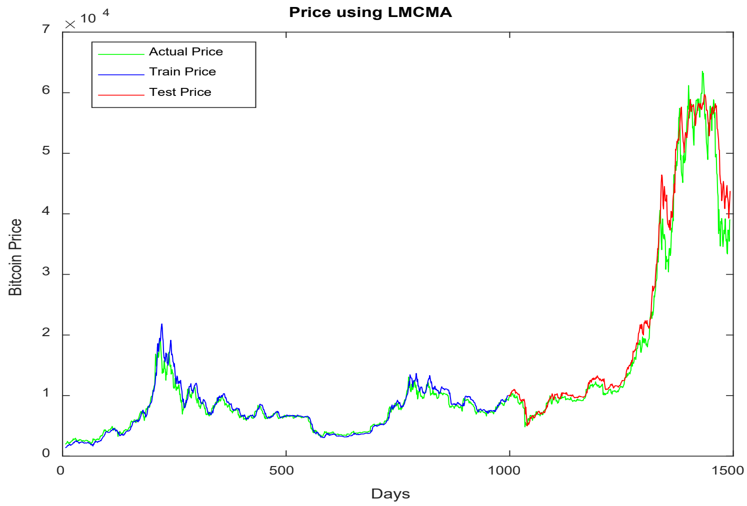

Figure 3. Bitcoin prices using LMCMA for BTC/USD dataset.

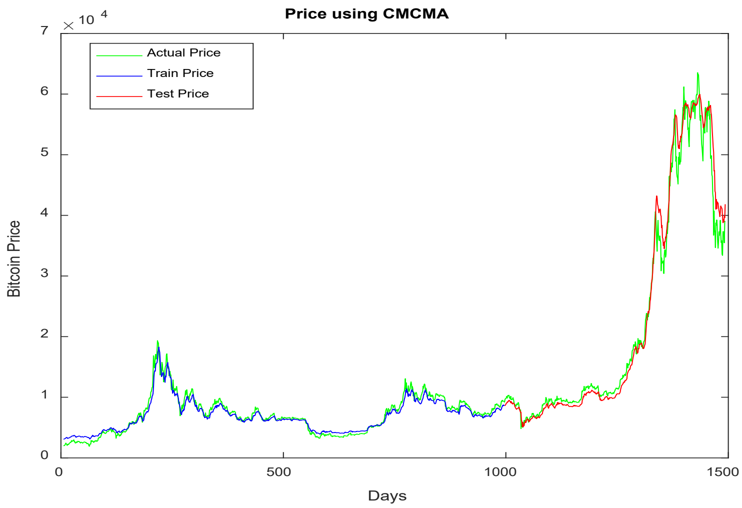

Figure 4. Bitcoin prices using CMCMA for BTC/USD dataset.

Similarly, Figures 5 and 6 provide a clear picture of the prediction capabilities of the CMCMA and CABCMA models, respectively, for the BTC/EUR dataset. In addition, the CMCMA model predicts Bitcoin prices with fewer errors than the CABCMA model.

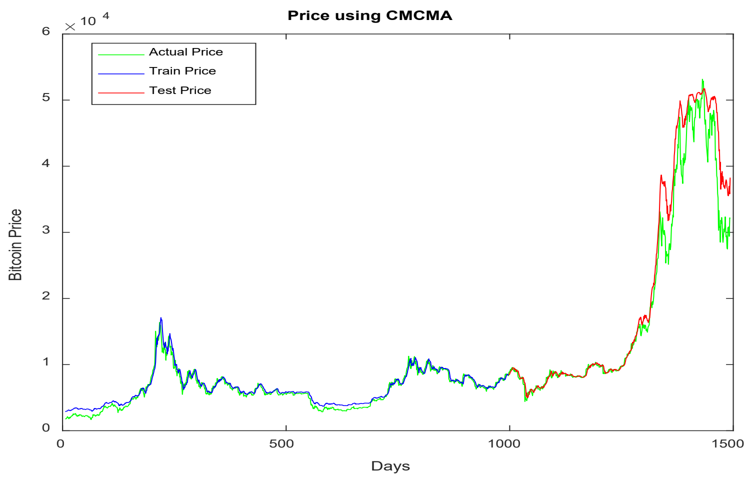

Figure 5. Bitcoin prices using CMCMA for BTC/EUR dataset.

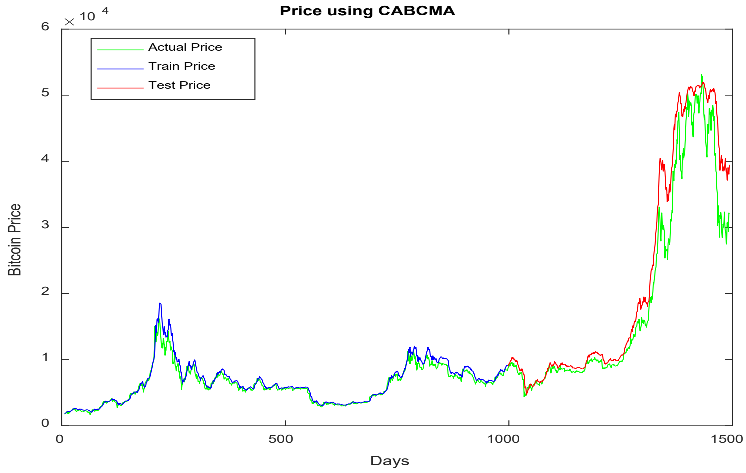

Figure 6. Bitcoin prices using CABCMA for BTC/EUR dataset.

Figures 7 and 8 show the actual price, the corresponding training and the testing prices generated by the CMCMA and LMCMA models, respectively, for the BTC/JPY dataset. Again, the CMCMA and LMCMA model approximate Bitcoin prices with very few intervals of over-prediction and under-prediction.

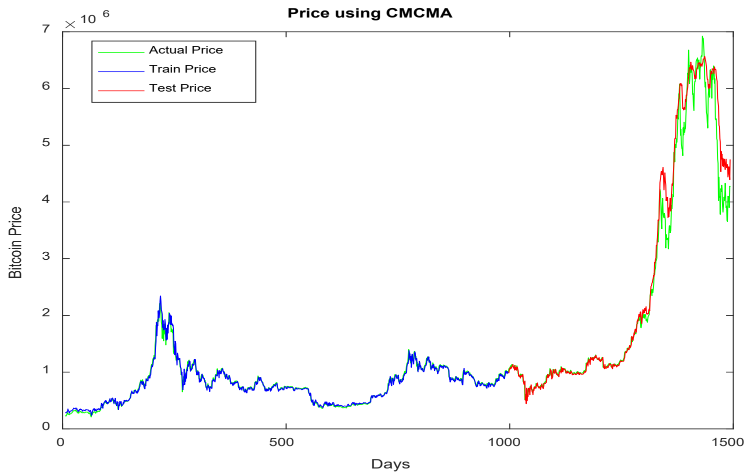

Figure 7. Bitcoin prices using CMCMA for BTC/JPY dataset.

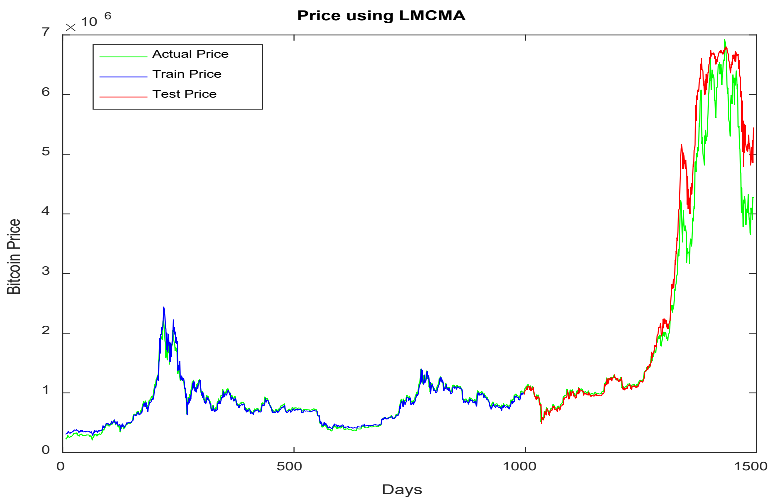

Figure 8. Bitcoin prices using LMCMA for BTC/JPY dataset.

In addition, to determine the learning capability of each model, the RMSE fitness value was plotted against the number of iterations for all models under study using the three datasets. Figures 9–11 show the RMSE plots for the BTC/USD, BTC/EUR, and BTC/JPY datasets, respectively. It is observed that the fitness value is better minimized by the MCMA models than by the other models.

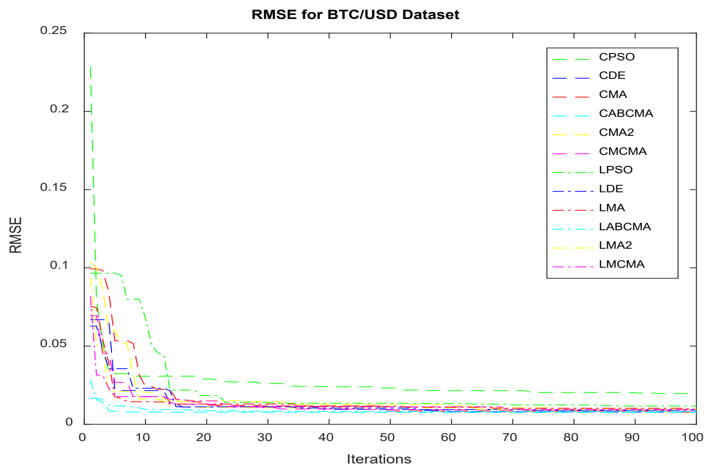

Figure 9. RMSE plot for BTC/USD dataset.

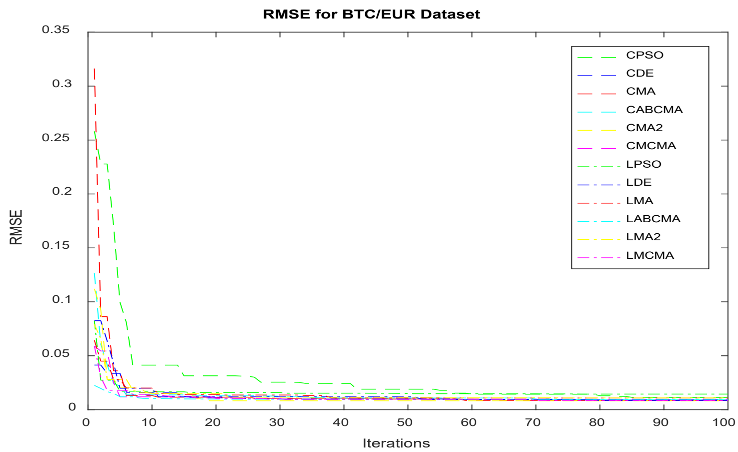

Figure 10. RMSE plot for BTC/EUR dataset.

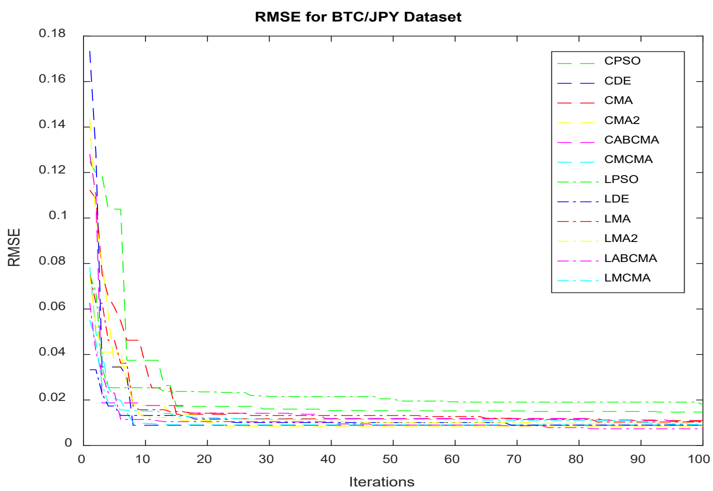

Figure 11. RMSE plot for BTC/JPY dataset.

6. Discussion

Bitcoin is recently being considered a safe-haven asset by some investors. However, the high volatility and decentralized governance are acting as the main hindrance to accepting Bitcoin as a safe asset. The stable prediction of Bitcoin price can help in persuading investors and stakeholders to consider Bitcoin as a prized possession. This can act as fuel to the positive socio-economic growth of the crypto-market and hence lead to sustainable development.

Recently, digital transformation is in full swing. Many industries and organizations are opting for a strong digital transformation. In this scenario, Bitcoin can play a very crucial role as it is a completely digital asset. Bitcoin is now available as Bitcoin Cash which supports digital transactions without any government or centralized control [36]. Additionally, blockchain is believed to transform industries for complete digitization [37]. Many industry giants are accepting Bitcoin as a means of payment for various digital transactions. Therefore, the prediction of Bitcoin prices can also aid in digital transformation.

In this study, a new MCMA algorithm is proposed for the efficient prediction of highly volatile daily Bitcoin prices. This algorithm is used in the CPBNN and LPBNN to generate two models for prediction. This study also includes a prediction analysis of ten other models using three different datasets. As the performance evaluation of these 12 models is less reliable if a single criterion is used, this study incorporated the TOPSIS framework to evaluate the models using 15 different criteria. It is visualized and inferred that the CMCMA model outperforms the other 11 models on two of the datasets, whereas the LMCMA model is the best predictor model for the remaining dataset. This study indicates that the prediction capability of the MCMA algorithm outperforms the other compared algorithms for this experimental setup, with a more promising multi-criteria performance analysis.

The results displayed in the tables and figures in this study indicate that the CMCMA model is a stable Bitcoin price prediction model in this scenario. Any change in input parameters has a subtle effect on its efficiency. The splitting of datasets into training and testing datasets also aids in the stability issues of the model by predicting prices for both datasets efficiently even if there is no common data between them.

The novelty of this research paper is compared with certain previously published papers in this subject area. It is shown in Table 13.

Table 13. Comparison of proposed work with other published work.

Reference No.	Method	Input Features	Metrics
[7]	Bayesian Neural Network	Blockchain Information	RMSE, MAPE
[8]	Classification tree-based model	124 technical indicators	Win-ratio, Loss-ratio
[12]	Deep Learning Method	40 determinant features	MAPE, RMSE, DA
This Paper	Polynomial-based Neural Network	Closing prices from 14 May 2017 to 14 June 2021	15 error metrics with TOPSIS ranking

However, this study has certain limitations. This study only focuses on the closing prices of Bitcoin. As Bitcoin is a stochastic currency that is highly non-stationary in its behavior, a single closing price prediction may not be sufficient for supporting ESG issues. It can be further extended for maximum, minimum, and direction prediction. As shown in Figures 3–8, the training prices predicted by the best two models in this study are coinciding with the actual prices. However, the testing prices were predicted inefficiently until around day 1300. After day 1300, the test prices are less coinciding with the actual prices. This may be the result of the ongoing COVID-19 pandemic. It can be further analyzed by using a post-COVID-19 dataset. This study relies on parameter tuning of many parameters in MCMA. The parameters were tuned and the best values were used for prediction with positive stability concerns. Still, more effective parameter tuning can give better test results. This study has not been implemented by any previously designed Bitcoin prediction models. Therefore, the efficiency of our proposed model is yet to be compared with other Bitcoin predictors.

7. Conclusions

Bitcoin is emerging as a promising cryptocurrency. This study proposes the use of a new MCMA algorithm as a way to capture the dynamic changes in Bitcoin prices and hence, supports the conclusion that Bitcoin is a safe-haven asset in the near future. A PBNN-based model has been considered for Bitcoin price prediction. To analyze the capability of the network, two different networks, that is, CPBNN and LPBNN are used. A new MCMA learning algorithm is proposed for the Bitcoin price prediction. Input weights are assigned randomly and the final output weight for both networks are calculated using six learning algorithms, that is, PSO, DE, MA, MA2, ABCMA, and MCMA during the training phase. Empirical analysis of the 12 predictors (six for each network) is done for three different datasets, that is, BTC/USD, BTC/EUR, and BTC/JPY. An MCDM method known as TOPSIS has been executed to rank the twelve models based on the fifteen performance measures during testing. The MCMA learning approach proposed here showed promising results in both networks, that is, CPBNN and LPBNN, for all three datasets under consideration. The prediction capability of the top two models for each dataset is visualized using the plot of the actual Bitcoin price and predicted Bitcoin price during training and testing. To visualize the learning capabilities of each model during training, the minimizing process of RMSE fitness value is also shown.

This study can be used to further examine the learning capabilities of the proposed MCMA algorithm using other networks. It can be further evaluated against various technical indicators in the Bitcoin dataset for more accurate prediction. This model successfully predicts most prices, but there is a case of over-prediction in the most recent prices. This can be further analyzed using other models as well as other pre-processing techniques. This model can be used to predict weekly and monthly prices to provide better insight into the volatile features of Bitcoin. The proposed algorithm can also be used to analyze dynamic changes in sources other than Bitcoin datasets. The Bitcoin prices considered in this study use prices till June 2021. Hence, it can be further utilized in a more recent study on volatile price prediction techniques.

Author Contributions: Conceptualization, S.M.; methodology, S.M.; software, S.M.; validation, S.M.; formal analysis, S.M.; investigation, S.M.; resources, S.M.; data curation, S.M.; writing—original draft preparation, S.M.; writing—review and editing, R.D.; visualization, S.M.; supervision, R.D.; project administration, R.D.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://www.investing.com/crypto/bitcoin/historical-data (accessed on 15 June 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 24 April 2021).
2. Harvey, C.R. Cryptofinance. Available online: https://ssrn.com/abstract=2438299 (accessed on 30 June 2021).
3. Bouri, E.; Molnár, P.; Azzi, G.; Roubaud, D.; Hagfors, L.I. On the Hedge and Safe Haven Properties of Bitcoin: Is It Really More than a Diversifier? *Financ. Res. Lett.* **2017**, *20*, 192–198. [CrossRef]
4. Smales, L.A. Bitcoin as a Safe Haven: Is It Even Worth Considering? *Financ. Res. Lett.* **2019**, *30*, 385–393. [CrossRef]
5. Harvey, C.R. Bitcoin Myths and Facts. Available online: http://ssrn.com/abstract=2479670 (accessed on 28 August 2021).

6. Boguslavsky, D.V.; Sharova, N.P.; Sharov, K.S. Cryptocurrency as Epidemiologically Safe Means of Transactions: Diminishing Risk of SARS-CoV-2 Spread. *Mathematics* **2021**, *9*, 3263. [CrossRef]
7. Jang, H.; Lee, J. An Empirical Study on Modeling and Prediction of Bitcoin Prices with Bayesian Neural Networks Based on Blockchain Information. *IEEE Access* **2018**, *6*, 5427–5437. [CrossRef]
8. Huang, J.-Z.; Huang, W.; Ni, J. Predicting Bitcoin Returns Using High-Dimensional Technical Indicators. *J. Financ. Data Sci.* **2019**, *5*, 140–155. [CrossRef]
9. Mallqui, D.C.A.; Fernandes, R.A.S. Predicting the Direction, Maximum, Minimum and Closing Prices of Daily Bitcoin Exchange Rate Using Machine Learning Techniques. *Appl. Soft Comput.* **2019**, *75*, 596–606. [CrossRef]
10. Peng, Y.; Albuquerque, P.H.; Camboium de Sá, J.M.; Padula, A.J.; Montenegro, M.R. The Best of Two Worlds: Forecasting High Frequency Volatility for Cryptocurrencies and Traditional Currencies with Support Vector Regression. *Expert Syst. Appl.* **2018**, *97*, 177–192. [CrossRef]
11. Aggarwal, D.; Chandrasekaran, S.; Annamalai, B. A Complete Empirical Ensemble Mode Decomposition and Support Vector Machine-Based Approach to Predict Bitcoin Prices. *J. Behav. Exp. Financ.* **2020**, *27*, 100335. [CrossRef]
12. Liu, M.; Li, G.; Li, J.; Zhu, X.; Yao, Y. Forecasting the price of Bitcoin using deep learning. *Financ. Res. Lett.* **2021**, *40*, 101755. [CrossRef]
13. Troumbis, I.A.; Tsekouras, G.E.; Tsimikas, J.; Kalloniatis, C.; Haralambopoulos, D. A Chebyshev Polynomial Feedforward Neural Network Trained by Differential Evolution and Its Application in Environmental Case Studies. *Environ. Model. Softw.* **2020**, *126*, 104663. [CrossRef]
14. Mohanty, S.; Dash, R. A Flower Pollination Algorithm Based Chebyshev Polynomial Neural Network for Net Asset Value Prediction. *Evol. Intell.* **2021**. [CrossRef]
15. Zhou, Y.; Zhu, Q.; Huang, H. Prediction of Acute Hypotensive Episode in ICU Using Chebyshev Neural Network. *J. Softw.* **2013**, *8*, 1923–1931. [CrossRef]
16. Dash, R. An Improved Shuffled Frog Leaping Algorithm Based Evolutionary Framework for Currency Exchange Rate Prediction. *Phys. A Stat. Mech. Appl.* **2017**, *486*, 782–796. [CrossRef]
17. Dash, R. DECPNN: A Hybrid Stock Predictor Model Using Differential Evolution and Chebyshev Polynomial Neural Network. *Intell. Decis. Technol.* **2018**, *12*, 93–104. [CrossRef]
18. Nanda, S.K.; Tripathy, D.P.; Nayak, S.K.; Mohapatra, S. Prediction of Rainfall in India Using Artificial Neural Network (ANN) Models. *Int. J. Intell. Syst. Appl.* **2013**, *5*, 1–22. [CrossRef]
19. Liu, F.; Wang, J. Fluctuation Prediction of Stock Market Index by Legendre Neural Network with Random Time Strength Function. *Neurocomputing* **2012**, *83*, 12–21. [CrossRef]
20. Dash, R. Performance Analysis of an Evolutionary Recurrent Legendre Polynomial Neural Network in Application to FOREX Prediction. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *32*, 1000–1011. [CrossRef]
21. Afifi, A.; Zanaty, E.A. Generalized Legendre Polynomials for Support Vector Machines (SVMS) Classification. *Int. J. Netw. Secur. Appl.* **2019**, *11*, 87–104. [CrossRef]
22. Dash, R.; Dash, P.K. MDHS–LPNN: A Hybrid FOREX Predictor Model using A Legendre Polynomial Neural Network with A Modified Differential Harmony Search Technique. In *Handbook of Neural Computation*; Samui, P., Sekhar, S., Balas, V.E., Eds.; Academic Press: London, UK; San Diego, CA, USA, 2017; pp. 459–486.
23. Peng, Z.; Yin, H.; Pan, A.; Zhao, Y. Chaotic Monkey Algorithm Based Optimal Sensor Placement. *Int. J. Control Autom.* **2016**, *9*, 423–434. [CrossRef]
24. Zhou, Y.; Chen, X.; Zhou, G. An Improved Monkey Algorithm for A 0–1 Knapsack Problem. *Appl. Soft Comput.* **2016**, *38*, 817–830. [CrossRef]
25. Marichelvam, M.K.; Tosun, O.; Geetha, M. Hybrid Monkey Search Algorithm for Flow Shop Scheduling Problem under Makespan and Total Flow Time. *Appl. Soft Comput.* **2017**, *55*, 82–92. [CrossRef]
26. Abiyev, R.H.; Tunay, M. Experimental Study of Specific Benchmarking Functions for Modified Monkey Algorithm. *Procedia Comput. Sci.* **2016**, *102*, 595–602. [CrossRef]
27. Zheng, L. An Improved Monkey Algorithm with Dynamic Adaptation. *Appl. Math. Comput.* **2013**, *222*, 645–657. [CrossRef]
28. Chen, X.; Zhou, Y.; Luo, Q. A Hybrid Monkey Search Algorithm for Clustering Analysis. *Sci. World J.* **2014**, *2014*, 938239. [CrossRef]
29. Sun, G.; Lan, Y.; Zhao, R. Self-Organizing Hierarchical Monkey Algorithm with Time-Varying Parameter. *Neural Comput. Appl.* **2019**, *8*, 3245–3263. [CrossRef]
30. Zulqarnain, R.M.; Saeed, M.; Ahmad, N.; Dayan, F.; Ahmad, B. Application of TOPSIS Method for Decision Making. *Int. J. Sci. Res. Math. Stat. Sci.* **2020**, *7*, 76–81.
31. Cocis, A.-D.; Batrancea, L.; Tulai, H. The link between corporate reputation and financial performance and equilibrium within the airline industry. *Mathematics* **2021**, *9*, 2150. [CrossRef]
32. Batrancea, L.M.; Nichita, A.; Cocis, A.-D. Financial performance and sustainable corporate reputation: Empirical evidence from the airline business. *Sustainability* **2022**, *14*, 13567. [CrossRef]
33. Sabaghi, M.; Mascle, C.; Baptiste, P. Application of DOE-TOPSIS Technique in decision-making problem. *IFAC-PapersOnLine* **2015**, *48*, 773–777. [CrossRef]

34. Samal, S.; Dash, R. A TOPSIS-ELM Framework for Stock Index Price Movement Prediction. *Intell. Decis. Technol.* **2021**, *15*, 201–220. [CrossRef]
35. Dash, R.; Samal, S.; Dash, R.; Rautray, R. An Integrated TOPSIS Crow Search Based Classifier Ensemble: In Application to Stock Index Price Movement Prediction. *Appl. Soft Comput.* **2019**, *85*, 105784. [CrossRef]
36. Bruijl, G.H.T. Adopting Bitcoin as a Digital Currency through Digital Transformation. Available online: https://ssrn.com/abstract=3030499 (accessed on 4 November 2022).
37. Hossain, S.A. Blockchain computing: Prospects and challenges for digital transformation. In Proceedings of the 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), Noida, India, 20–22 September 2017; IEEE: Noida, India, 2017; pp. 61–65.