Enhancing Bitcoin Price Prediction with Evolutionary Radial Bias Function Networks

IJIMSR, Vol. 2, No. 1, (2024) 24.2.1.016

***Sudersan Behera**
Associate Professor
Sphoorthy Engineering College,
Hyderabad, 501510, India
Email: sbehera.04@gmail.com

P M Suresh
Assistant Professor
Sphoorthy Engineering College,
Hyderabad, 501510, India
Email: sureshpm.networks@gmail.com

***Corresponding author**

Received 16th Mar 2024; Accepted 4th May 2024

KEYWORDS: Time Series Forecasting, Radial Bias Function Network, Fireworks Algorithm

Enhancing Bitcoin Price Prediction with Evolutionary Radial Bias Function Networks

***Sudersan Behera**
Associate Professor
Sphoorthy Engineering College,
Hyderabad, 501510, India
Email: sbehera.04@gmail.com

P M Suresh
Assistant Professor
Sphoorthy Engineering College,
Hyderabad, 501510, India
Email: sureshpm.networks@gmail.com

ABSTRACT

A semi-parametric evolutionary method called the Fireworks method (FWA) and a radial bias function neural network (RBFN) are combined in this study to make a new type of network called RBFN-FWA. Effectively adjusting the RBFN's biases and weights is the goal of integrating FWA. We next used this proposed methodology to predict how Bitcoin's price will go in the future. The solo RBFN was trained using various optimization techniques such as GA, PSO, and GD for comparison purposes. This resulted in three new models: RBFN-GA, RBFN-PSO, and RBFN-GD. We also use all the other models that can do the same thing. Utilizing error metrics like Mean Absolute Percentage Error (MAPE) and Normalized Mean Squared Error (NMSE), we evaluated the models' performance. In terms of NMSE and MAPE, the experimental findings show that RBFNN-FWA is the best comparison model, demonstrating its higher predictive potential.

KEYWORDS: Time Series Forecasting, Radial Bias Function Network, Fireworks Algorithm

1. INTRODUCTION

The stock market, banks, and the conversion of assets into digital currency are all parts of the financial market, which in turn affects the economy. The term "trading" refers to the buying and selling of a wide range of financial products. Due diligence is essential before allocating funds to the stock market, as financial wagers carry a significant degree of risk. The unpredictable behavior of the stock market has piqued the interest of researchers and financial organizations in developing trustworthy prediction systems. Financial time series trends reflect the general movement of stock prices and are particularly difficult to analyze due to the market's nonlinearity, volatility, and lack of stationarity. In order to make reliable predictions, stakeholders and investors need to understand the factors that impact market behavior, such as investor sentiment, political manoeuvring, economic circumstances, and commodity prices. Data analysis is essential for overcoming many of the challenges encountered in financial market research. Financial forecasting has seen extensive usage of AI technologies such as sentiment analysis, machine learning, and deep learning. With the help of AI techniques like multilayer perceptrons [1], support vector regression (SVR) [2], recurrent neural networks (RNN) [3], and deep learning protocols [4], tasks like stock prediction can now be done automatically. More and more people are turning to hybrid optimization neural networks (HONNs) [5] because of their great learning capabilities, fast computation rates, and massive memory capacities. These properties enhance the precision of predictions and classifications [6]. Also, in order to improve efficiency and performance in the financial markets, a number of metaheuristic algorithms have been employed to deal with challenging refining problems. They consist of several algorithms, such as PSO [11], TLBO [12], CRO [13], GA [8], FWA [9], FFA [10], PSO [11], and many more. These algorithms offer versatile methods to enhance forecast accuracy and optimize trading strategies in dynamic markets. Here is how the rest of the article is structured: In Section 2, we delve into the existing background study. In Section 3, we offer a comprehensive explanation of the materials and methods used in this study. In Section 4, we learn about the dataset and how the model was trained. In Section 5, we explain the simulation setup and analyze the results. Finally, in Section 6, we wrap up with some final thoughts. The bibliography is listed at the conclusion.

2. LITERATURE SURVEY

Although HONNs are used in many other financial applications, our study is focused on their exceptional effectiveness in financial market prediction. For financial time series forecasts, HONNs and variants are commonly used because of their fast learning, high generalization capacity, and minimal complexity structure. These models have been found to be more robust and dependable than other artificial neural network (ANN) models. Any trader serious about making money investing needs a solid plan for making predictions and some solid rules to follow. The combination of non-linearity and market volatility makes it difficult to build a dependable predictive model. Tolerance for error, capability to identify market anomalies and uncertainty, coupled with high accuracy through mistake minimization or elimination are the hallmarks of a trustworthy financial prediction model. A trustworthy prediction model also has the capacity to adjust to different training procedures to get the best results. According to Nayak et al. [14], a pi-sigma HONN was suggested for stock index forecasting. They also devised an improved neural network model for computational efficiency in financial time series forecasting [16]. Dash et al. [15] created a CSA-based PSNN predictor for gold price prediction, whereas Sahoo et al. [29] used a PSNN in conjunction with MOGA to forecast stock market prices. The SDE technique was also used to train a PSNN by Dash et al. [17].In [27], the author examined how well a HONN with ISFLA predicted currency exchange rates. In their study, Nayak et al. [18] presented and evaluated a new HONN-based FFA that draws inspiration from nature. The PSO-GA-based PSNN and the new HBMO with HONN proposed by Nayak & Naik [19, 22] and Nayak et al. [19] both use the standard backpropagation GD Learning to sort things into groups. Using an IPSO and a higher-order PSNN, Kanungo et al. [23] developed and tested a non-linear classifier. Nayak et al. [24] optimized a HONN using the TLBO algorithm, and they also proposed a PSO-based HONN for classification [25]. For fuzzy time series forecasting, Pattanayak et al. [21] presented a unique hybrid DE-PSNN, while Nie and Deng [20] examined the convergence of a GA-based PSNN. They went a step farther and suggested a Pi-Sigma HONN hybrid CRO for the identical objective [28]. Last but not least, a PSNN was used to examine the identical time series by Bas et al. [30]. Similarly, RBFN, another kind of HONN, has helped with a number of financial forecasting tasks, including time series forecasting [31], corrosion rate prediction [32], wear prediction in drilling [33], and network traffic flow forecasting [34].

The literature review highlights the prevalence of hybrid evolutionary models in financial forecasting, predominantly relying on parametric evolutionary-based approaches. However, less attention has been paid to the utilization of non-parametric evolutionary algorithms. This gap motivates our research, where we propose to integrate a semi-parametric algorithm such as FWA with RBFN to optimize the weights and biases of RBFN, yielding a novel model termed RBFN-FWA. Subsequently, we apply utilizing a hybrid approach for forecasting the closing price of Bitcoin, addressing the aforementioned gap in the literature.

3. MATERIALS AND METHOD

In this section, we will delve into GA, PSO, FWA, and GD algorithms, alongside the RBFN forecasting model.

3.1 Genetic Algorithm

Taking cues from Darwin's idea of natural selection, Holland unveiled the genetic algorithm in 1975 [8]. This method mimics natural selection by randomly mixing solutions and giving preference to the best parents. The genetic algorithm guides the population towards an ideal solution over subsequent generations by randomly selecting members of the current population as parents in every generation. To learn about the widespread use of genetic algorithms for forecasting financial time series, interested readers can peruse a number of studies in the literature [20, 35]. In a standard genetic algorithm, there are four steps:

a. Initial Population: A group of chromosomes, one for each possible solution to an issue, constitutes the starting point of the procedure. These chromosomes' genes determine their traits.

b. Selection: The probability of a chromosome being inherited from one generation to the next is directly proportional to its fitness level.

Crucial to the genetic algorithm is the crossover phase. In order for an organism to develop into a new species, the DNA of both parents must be transferred during mating, and a random crossover point is randomly chosen from each pair of genes. In order to improve algorithmic exploitation, the crossover operator investigates the region around a chromosome.

c. Mutation: Exploration can be facilitated by the mutation operator if certain genes in newly created progeny undergo mutation. Using chromosomes to define solutions is a cornerstone of genetic algorithm modeling. Figure 1 depicts the chromosome structure, the single-point crossover operation, and the mutation method used in genetic algorithms.

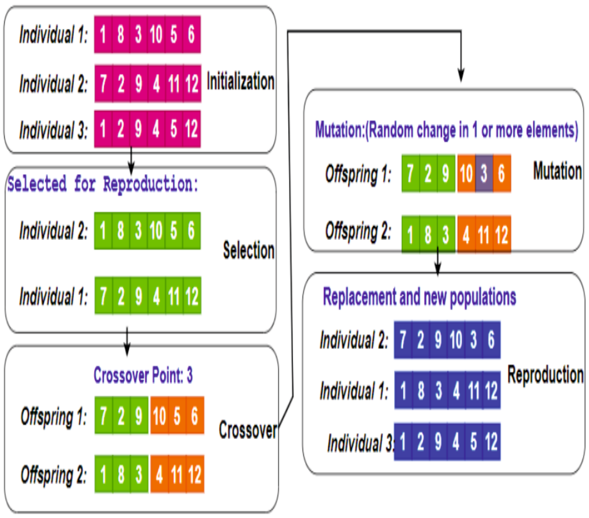

Figure 1 GA Operations

3.2 Particle Swarm Optimization (PSO)

The PSO algorithm is among the most influential swarm intelligence and intelligent optimization approaches. The PSO method, first suggested and created in 1995 [11] by Russell C. Eberhart and James Kennedy, the concept aims to emulate the social behaviors of animals, resembling a flock of birds searching for food. This program mirrors human communication and information exchange. PSO has proven effective in resolving numerous optimization challenges, either independently or in conjunction with other contemporary techniques. This approach finds the best solution by using agents, called particles, whose paths are affected by both random and predictable factors. When in reality, the group's "best" position and each particle's best position are influencing how particles move. You may see the PSO flowchart in Figure 2. The literature contains pertinent publications [36–37] for anyone interested in the use of PSO for FTS prediction.

Figure 2 Working Principles of PSO

3.3 Gradient Descent (GD) Learning

GD serves as an optimization technique employed to minimize the cost function in machine learning and deep learning models. Its principal objective lies in discovering the optimal parameters (weights and biases) of a model, consequently reducing the error or loss function. The operational framework of the GD Learning [38] is depicted in Figure 3.

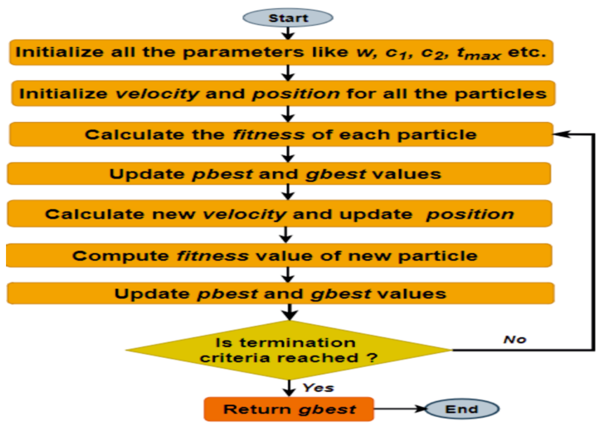

Figure 3 Working principles of GDL

3.4 Radial Basis Function Network (RBFN)

RBFN [39] stands out among artificial neural networks with its distinctive two-layer design. Neuron activation within the hidden layer depends on the distance from input-space centers, employing radial basis functions for this purpose. By consistently utilizing the radial activation function, the network demonstrates exceptional performance in tasks such as function approximation and pattern recognition. Utilizing RBFNs [40] for financial time series prediction serves as a notable application, and modifying their architecture can potentially enhance their performance in specialized domains. This network proves valuable across various contexts because of its ability to cluster input patterns based on the centers of radial basis functions, thereby enabling it to discern complex correlations within the data. An architecture of RBFN forecasting model given in Figure 4. The key component of RBFN is the radial basis function, typically Gaussian, which measures the similarity between input data and reference points called centroid. The Gaussian RBF function φ_j for the j^{th} neuron is defined as:

$$\varphi_j(x) = \exp^{-\sum_{j=1}^{m}(x-p_j)^2 / -2r_j^2} \qquad (1)$$

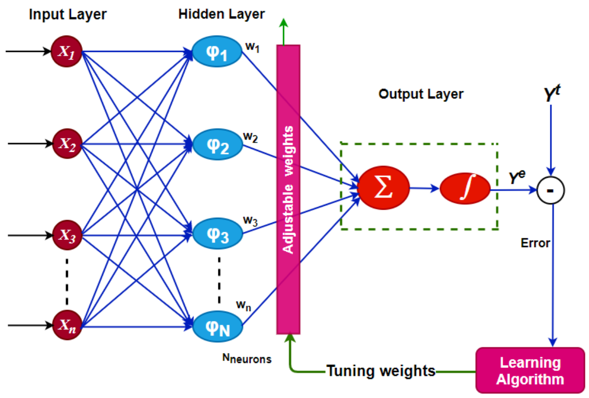

Figure 4 Architecture of RBFN Forecasting model

Where x is the input vector, p_j is the centroid vector of the j^{th} neuron, and r_j is the width parameter (spread) of the j^{th} neuron's RBF.

$$Y^e = f(w,x) = \sum_{i=1}^{N} w_i \varphi_i(x) + bias \qquad (2)$$

Where, w_i are the weights connecting hidden layer neurons to the output and Y^e is the output of the network.

3.5 Fireworks Algorithm (FWA)

In 2010 [41], Tan and Zhu presented the Fireworks Algorithm (FWA), which stands for an evolutionary algorithm that was newly devised and is based on swarm intelligence. The process begins with the selection of a fixed number of spots in the solution space, much like fireworks. As a result of their explosive mechanism, these pyrotechnics produce a shower of sparks. After iteratively improving the solution's quality, the best solutions derived from fireworks and sparks are selected to be the fireworks for the subsequent generation. Thanks to its impressive global search and information usage capabilities, FWA has garnered significant attention from researchers and has shown great performance on a variety of real-world challenges. One way to think about an explosion in a localized setting is as a search. In Figure 5, we can see the FWA operating concept. The amount of sparks C_i can be expressed as follows if k_i is a explosive and M_i is the amplitude of an outburst:

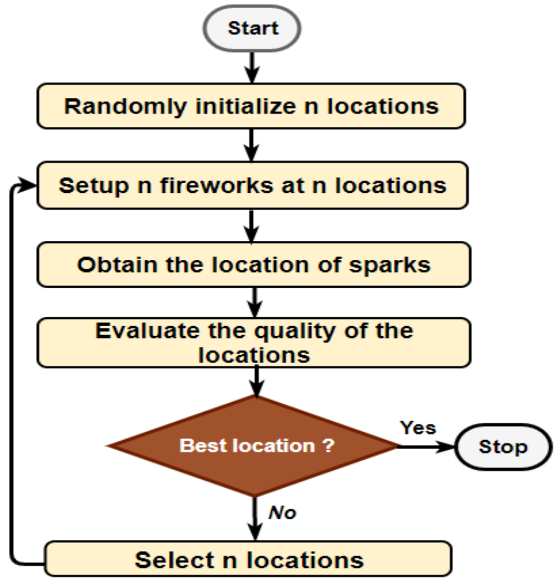

Figure 5 Working principle of FWA

$$M_i = \vec{M} \cdot \frac{f(k_i) - f_{\min} + \varepsilon}{\sum_{j=1}^{p}(f(k_j) - f_{\min} + \varepsilon} \qquad (3)$$

$$C_i = \frac{m.f_{\max} - f(k_i) + \varepsilon}{\sum_{j=1}^{p}(f(k_{\max} - f(k_i)) + \varepsilon} \qquad (4)$$

In Eq. (3), the largest possible explosion amplitude is denoted by $f_{\min}$ the least and most objective functions among P fireworks are denoted by $f_{\min}$ and $f_{\max}$, respectively; by m, the parameters that determine the sparks produced by a firework; and by ε, a constant that prevents zero division.

4 ABOUT THE DATASET AND TRAINING RBFN MODEL

We collected daily Bitcoin data from www. yahoofinance. com spanning from January 2, 2020, to January 2, 2024. The datasets underwent pre-processing using sigmoid normalization [42]. Figure 6 illustrates the fundamental statistical properties of the Bitcoin closing price, while Figure 7 depicts the historical trend of Bitcoin's closing prices. Among the various available features, we selected Open, Close, Adj Close, High, and Low as inputs for the RBFN model.

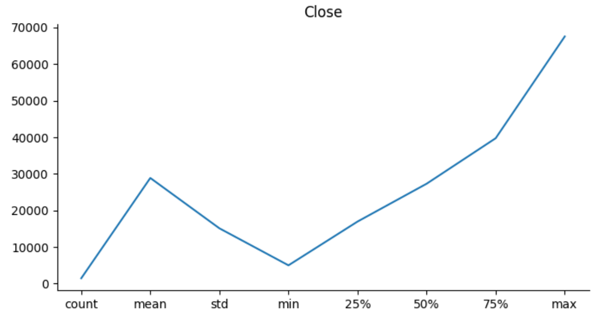

Figure 6 Statistical Properties of Close Price of Bitcoin

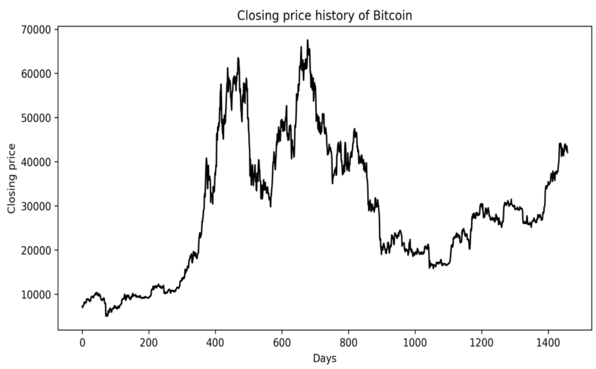

Figure 7 Close Price history of Bitcoin

Employing a sliding window [42] procedure with a window size of 3, we divided these features into training and testing datasets, maintaining a 60:40 split ratio between the two sets of data. A sliding widow of size 3 is depicted in figure 8.

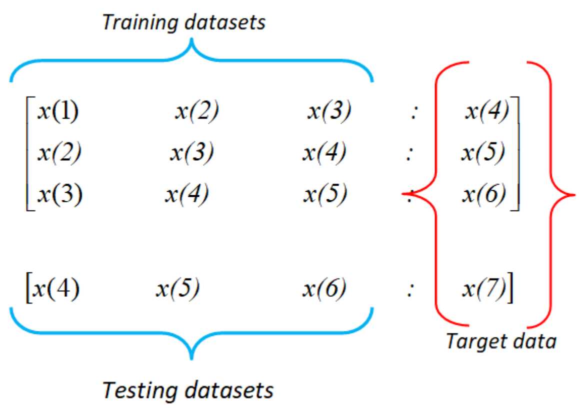

Figure 8 Sliding window of size 3

Our prediction model underwent training and evaluation using these methods. The proposed RBFN metaheuristic technique is founded on a single-hidden-layer model. FWA, GA, PSO, and GD were individually employed to determine the weights of hidden-output neurons and the bias vectors connecting them. The entire training process of RBFN using FWA is visually depicted in Figure 9. Suppose that the input vector $i[n]=[i_1, i_2, i_3, \ldots, i_n]$ represents the modified closing prices. Suppose that the input vector $i[n]$ is coupled to a weight vector

$$w[n]=[w_{11}, w_{12}, \ldots, w_{nm}]$$

for each summing unit neuron. The output neuron of the model receives the final closing price after the model is inputted with a series of patterns $i[n]$ The model's predicted output, Y^e, is very near to the goal value, Y^t, when fed the input data. Here is the process for deriving the error signal e(n)

$$e(n) = Y^t - Y^e$$

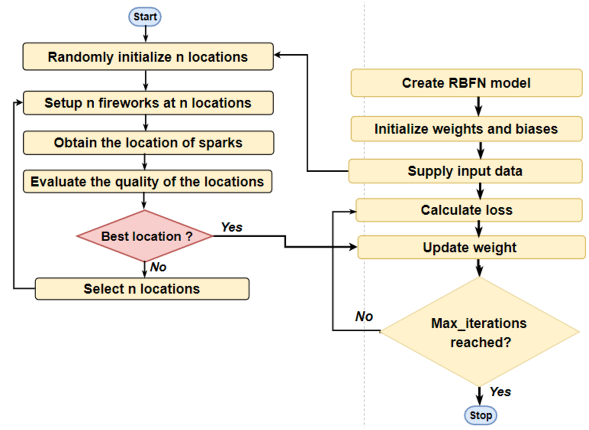

Figure 9 Training RBFN using FWA

Algorithm 1 delineates a top-tier methodology for RBFN model training via an evolutionary algorithm. Researchers keen on employing this technique may choose any single evolutionary algorithm from those outlined above for their training endeavors.

Algorithm-1: Procedure to train RBFN using evolutionary approach

1. Adjust evolutionary parameters such as mutation rate, crossover rate, and others.
2. Initialize the population.
3. Generate training and testing data using the sliding window method.
4. While there is remaining test data:
5. Set a counter variable (counter) to 1.
6. Normalize the data using the Sigmoid function.
7. **Training Phase:**
8. Initialize an iteration variable (iteration) to 0.
9. While iteration is less than or equal to a predefined maximum iteration limit:
10. Provide the training data and an individual from the evolutionary algorithm to the RBFN model.
11. Calculate hidden layer signals.
12. Compute the output at the output neuron.
13. Use metaheuristic operators to update weights and thresholds.

14. Identify the best-fit individual and update the population accordingly.
15. Increment iteration by 1.
16. End the iteration loop.
17. **Testing Phase:**
18. Use the test data and the fittest individual from the population in the RBFN model.
19. Evaluate the RBFN output and calculate the error.
20. Increment counter by 1.
21. End the testing loop.
22. Calculate the average error over all test cases.
23. Set a small value for counter.

5 RESULT AND DISCUSSION

All models were designed using Google Colab in a Windows 10 environment. Each model was individually applied to forecast the closing price of Bitcoin. Initially, they were trained using training datasets, and once the trained model fixed optimal weights and biases, test datasets were applied. Every model was employed to predict the closing price over two different time horizons: 1-day ahead and 7-day ahead forecasting. We employed MAPE and NMSE metrics as defined by Behera et al. [42] to gauge the concert of each model. The testing performance of each model was evaluated in terms of Average MAPE and Average NMSE for 10 runs. Table 1 presents the error comparison for each model, and for visualization purposes, we calculated the 1-day-ahead average and 7-day-ahead average for each model. The pair comparison bar plot is depicted in Figure 10.

Table 1 Error Score Comparison

Mo	Error	Period	BTC -
RBFN - FWA	MAPE	1-day	0.49123
		7-day	0.58901
	NMSE	1-day	0.48992
		7-day	0.57851
RBFN - PSO	MAPE	1-day	0.50124
		7-day	0.59902
	NMSE	1-day	0.49993
		7-day	0.58852
RBFN - GA	MAPE	1-day	0.51125
		7-day	0.60903
	NMSE	1-day	0.50994
		7-day	0.62853
RBFN - GD	MAPE	1-day	0.62205
		7-day	0.71983
	NMSE	1-day	0.64074
		7-day	0.75933

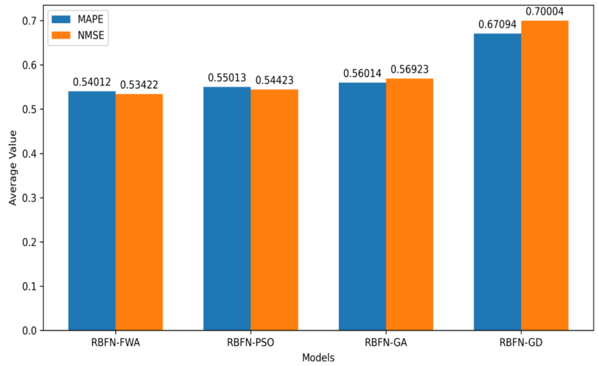

Figure 10 Average Errors for 1-Day and 7-Day Bitcoin Price Forecasts.

Analyzing the data presented in Table 1 and Figure 9 reveals a distinct trend: the RBFN model, trained with the semi-parametric algorithm FWA, consistently exhibits superior performance compared to its counterparts trained with parametric algorithms such as GA, PSO, and the conventional GD algorithm. Notably, the RBFN-FWA model outperforms competitive models, including RBFN-GA, RBFN-PSO, and RBFN-GD, in terms of both MAPE and NMSE values. This superiority holds true for both 1-day-ahead and 7-day-ahead forecasting horizons, emphasizing the robust forecasting capabilities of the RBFN-FWA model in the context of Bitcoin closing price predictions.

Additionally, the performance of RBFN-FWA can be scrutinized through the 1-day ahead forecasting plot generated for Bitcoin, as illustrated in Figure 10. This plot offers a detailed visualization of the model's predictive accuracy and its ability to capture the nuances of Bitcoin price movements over short-term intervals.

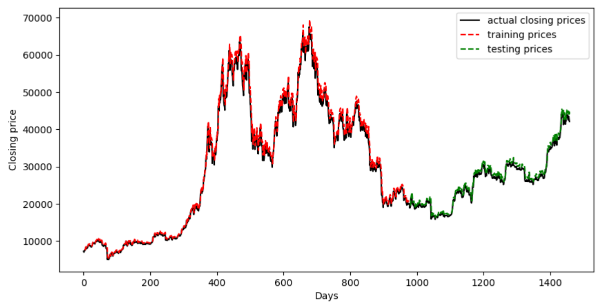

Figure 11 One-day ahead forecasting plot of RBFN-FWA for Bitcoin

6 CONCLUSIONS

In this research, we integrated a semi-parametric evolutionary algorithm, known as FWA, with an RBFN to create a hybrid network termed RBFN-FWA. FWA was utilized to effectively adjust the biases and weights of the RBFN. Subsequently, we applied this hybrid model to predict future price movements of Bitcoin. For comparison purposes, we trained the standalone RBFN using alternative optimization algorithms, including GA, PSO, and GDL, resulting in three additional models: RBFN-GA, RBFN-PSO, and RBFN-GDL. We evaluated the performance of all models using MAPE and NMSE metrics. The experimental results demonstrated that RBFN-FWA outperforms the other comparative models in terms of both NMSE and MAPE, indicating its superior predictive capability. In future research, we plan to explore the application of the FWA algorithm with other variants of ANNs to develop additional hybrid models. Furthermore, we intend to employ the RBFN-FWA model to forecast other financial time series data.

Statements and Declarations

1. Conflict of Interest: The authors have no obvious financial or other conflicts of interest.
2. Funding: Authors did not secure any financing, grants, or other kind of support to compile this work.
3. Data Availability: As previously stated by the authors, all data utilized in this study, including experimental data, may be found publically at https://www.yahoofinance.com.

Abbreviations

Multiple offspring genetic algorithm	(MOGA)
Shuffled Differential Evolution	(SDE)
Improved shuffled frog leaping algorithm	(ISFLA)
Firefly algorithm	(FFA)
Higher order neural network	(HONN)
Gradient descent	(GD)
Honey-Bees mating optimization	(HBMO)
Genetic Algorithm	(GA)
Differential Evolution	(DE)
Firework Algorithm	(FWA)
Chemical reaction optimization	(CRO)
Pi-sigma Neural Network	(PSNN)
Radial Bias Function Network	(RBFN)

REFERENCES

[1] Zong, X. (2020). MLP, CNN, LSTM and Hybrid SVM for Stock Index Forecasting Task to INDU and FTSE100. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3644034

[2] Zhu, B., Han, D., Wang, P., Wu, Zhanchi, Zhang, T., & Wei, Y.-M. (2017). Forecasting carbon price using empirical mode decomposition and evolutionary least squares support vector regression. Applied Energy. Elsevier, 191, 521–530. https://doi.org/10.1016/j.apenergy.2017.01.076

[3] Smith, C., & Jin, Yaochu. (2014). Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction. Neurocomputing. Elsevier, 143, 302–311. https://doi.org/10.1016/j.neucom.2014.05.062

[4] Kumar, S., Rastogi, U. A Comprehensive Review on the Advancement of High-Dimensional Neural Networks in Quaternionic Domain with Relevant Applications. Arch Computat Methods Eng (2023). https://doi.org/10.1007/s11831-023-09925-w

[5] Dr Perwej, A., Prof. (Dr.) K. P. (2018). Yadav, Prof. (Dr.) V Sood, Dr. Yusuf Perwej. An Evolutionary Approach to Bombay Stock Exchange Prediction with Deep Learning Technique, IOSR Journal of Business and Management, 20(12)

[6] Gourav, K., Sanjeev, J., & Pratap, S. U. (2021). Stock market forecasting using computational intelligence: A Survey. Computational Methods in Engineering, 28(3)

[7] Narayana, B. S., Nayak, S. C., & Kar, B. P. (2020). Design and performance evaluation of monarch butterfly optimization-based artificial neural networks for financial time series prediction, ICDECT, 2020. Springer

[8] Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press

[9] Tan, Y., & Zhu, Y. (2010). Fireworks algorithm for optimization. In Y. Tan, Y. Shi & K. C. Tan (Eds.), Advances in swarm intelligence lecture notes in computer science (pp. 355–364). Springer. https://doi.org/10.1007/978-3-642-13495-1_44

[10] Nayak, J., Naik, B., & Behera, H. S. (2015). A novel nature inspired firefly algorithm with higher order neural network: Performance analysis. Engineering Science and Technology, An International Journal. Elsevier

[11] Eberhart, R., & Kennedy, J. (October 4–6, 1995). A new optimizer using particle swarm theory. In Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS); Nagoya, Japan (pp. 39–43). https://doi.org/10.1109/MHS.1995.494215

[12] Nayak, J., Naik, B., & Behera, H. S. (2015). 'Optimizing a Higher Order Neural Network Through Teaching Learning Based Optimization Algorithm', computational intelligence in data mining. Advances in Intelligent Systems and Computing 410, International conference on CIDM, 1

[13] Alatas, B. (2011). ACROA: Artificial chemical reaction optimization algorithm for global optimization. Expert Systems with Applications, 38(10), 13170–13180.

https://doi.org/10.1016/j.eswa.2011.04.126

[14] Nayak, S. C., Misra, B. B., & Behera, H. S. (2015). A pi-sigma higher order neural network for stock index forecasting,. In Computational Intelligence in Data Mining, 2 (pp. 311–319). Springer

[15.] Dash, R., Routray, A., Dash, R., & Rautray, R. (2021). Designing an efficient predictor model using PSNN and crow search based optimization technique for gold price prediction. Intelligent Decision Technologies. IOS Press, 15(2), 281–289. https://doi.org/10.3233/IDT-200093

[16] Nayak, S. C., Kumar, K. V., & Satyanarayana, B. (2020). Towards designing A computationally efficient neural network model for improved financial time series forecasting. In Lecture notes in Electrical engineering, Advances in cybernetics,Cognition and Machine learning for communication Technologies, 605–619. https://doi.org/10.1007/978-981-15-3125-5_58

[17] Rajashree, D., Rasmita, R., & Rasmita, D. (2019). Utility of a Shuffled Differential Evolution algorithm in designing of a Pi-Sigma Neural Network based predictor model. Applied Computing and Informatics. Emerald

[18] Nayak, J., Naik, B., & Behera, H. S. (2015). A novel nature inspired firefly algorithm with higher order neural network: Performance analysis. Engineering Science and Technology, An International Journal. Elsevier.

[19] Nayak, J., Naik, B., & Behera, H. S. (2014). A hybrid PSO-GA based Pi sigma neural network (PSNN) with standard back propagation gradient descent learning for classification. In International conference on control, instrumentation, communication and computational technologies (ICCICCT), IEEE, 2014 (pp. 878–885). IEEE Publications. https://doi.org/10.1109/ICCICCT.2014.6993082

[20] Nie, Y., & Deng, W. (2008). A hybrid genetic learning algorithm for Pi–Sigma neural network and the analysis of its convergence. In Fourth international conference on natural computation, 3 (pp. 19–23). IEEE Publications. https://doi.org/10.1109/ICNC.2008.896

[21] Pattanayak, R. M., Behera, H. S., & Panigrahi, S. (2018). 'A Novel Hybrid Differential Evolution-PSNN for Fuzzy Time Series Forecasting', advances in intelligent systems and computing. In Proceedings of the International Conference on ICCIDM 2018.

[22] Nayak, J., & Naik, B. (2018). A novel Honey-Bees mating optimization approach with higher order neural network for classification. Journal of Classification, 35(3), 511–548. https://doi.org/10.1007/s00357-018-9270-1

[23]. Kanungo, D. P., Nayak, J., Naik, B., & Behera, H. S. (2016). Non-linear classification using higher order Pi-sigma neural network and improved particle swarm optimization: An experimental analysis. Advances in Intelligent Systems and Computing, 507–518. https://doi.org/10.1007/978-81-322-2731-1_48

[24] Nayak, J., Naik, B., & Behera, H. S. (2015). 'Optimizing a Higher Order Neural Network Through Teaching Learning Based Optimization Algorithm', computational intelligence in data mining. Advances in Intelligent Systems and Computing 410, International conference on CIDM, 1.

[25] Nayak, J., Naik, B., Behera, H. S., & Abraham, A. (2015). 'Particle Swarm Optimization Based Higher Order Neural Network for Classification', computational intelligence in data mining. Smart Innovation, Systems and Technologies, 1.

[26] Samir, L., Said, G., Nora, K., & Youcef, S. (2017). Improved Pi-Sigma Neural Network for nonlinear system identification, The 5th international Conference on Electrical Engineering—Boumerdès (ICEE-B). https://doi.org/10.1109/ICEE-B.2017.8192144

[27] Dash, R. (2018). Performance analysis of a higher order neural network with an improved shuffled frog leaping algorithm for currency exchange rate prediction. Applied Soft Computing, 67, 215–231. https://doi.org/10.1016/j.asoc.2018.02.043

[28] Pattanayak, R. M., Behera, H. S., & Panigrahi, S. (2020). A multi-step-ahead fuzzy time series forecasting by using hybrid chemical reaction optimization with Pi-Sigma higher-order Neural Network. In Computational intelligence in pattern recognition (pp. 1029–1041). https://doi.org/10.1007/978-981-13-9042-5_88

[29] Sahoo, S., Mohanty, S. K., & Pradhan, S. K. (2019). Combining PI Sigma neural network with multiple offspring genetic algorithm for stock market price prediction. International Journal of Engineering and Advanced Technology, ISSN: 2249-8958, 9(1), 6934–6939. https://doi.org/10.35940/ijeat.A2103.109119

[30]. Bas, E., Grosan, C., Egrioglu, E., & Yolcu, U. (2018). High order fuzzy time series method based on pi-sigma neural network. Engineering Applications of Artificial Intelligence. Elsevier, 72, 350–356. https://doi.org/10.1016/j.engappai.2018.04.017

[31] Satoshi Kitayama, & Kohei Saito. (2022, June). Time Series Prediction Using Radial Basis Function Network with Transformation of Training Data and Its Applications. Automatic Control and Computer Sciences, 56(3), 239–252. https://doi.org/10.3103/s0146411622030026

[32] Liu, Y., Han, X., Kang, L., & Li, Q. (2022, June). Prediction of Corrosion Rates of Ni-TiN composite coating using a Radial Basis Function Neural Network. International Journal of Electrochemical Science, 17(6), 220626. https://doi.org/10.20964/2022.06.15

[33] Panda, S., Chakraborty, D., & Pal, S. (2008, March). Flank wear prediction in drilling using back propagation neural network and radial basis function network. Applied Soft Computing, 8(2), 858–871. https://doi.org/10.1016/j.asoc.2007.07.003

[34] Guo, T., Lan, J. L., Li, Y. F., & Jiang, Y. M. (2014, February 25). Network Traffic Prediction with Radial Basis Function Neural Network Based on Quantum Adaptive Particle Swarm Optimization. Journal of Electronics & Information Technology, 35(9), 2220–2226. https://doi.org/10.3724/sp.j.1146.2012.01343

[35] Waheeb, W., & Ghazali, R. (2019). A new genetically optimized tensor product functional link neural network: An application to the daily exchange rate forecasting. Evolutionary Intelligence. Springer, 12(4), 593–608. https://doi.org/10.1007/s12065-019-00261-2

[36] Shamshirband, S., Hadipoor, M., Baghban, A., Mosavi, A., Bukor, J., & Várkonyi-Kóczy, A. R. (2019). Developing an ANFIS-PSO model to predict mercury emissions in combustion flue gases. Mathematics. MDPI, 7(10). https://doi.org/10.3390/math7100965

[37] Zhao, L., & Yang, Y. (2009). PSO-based single multiplicative neuron model for time series prediction. Expert Systems with Applications, 36(2), 2805–2812. https://doi.org/10.1016/j.eswa.2008.01.061

[38] Amari, S. I. (1993, June). Backpropagation and stochastic gradient descent method. Neurocomputing, 5(4–5), 185–196. https://doi.org/10.1016/0925-2312(93)90006-o

[39] Chen, D., & Han, W. (2013, February 23). Prediction of multivariate chaotic time series via radial basis function neural network. Complexity, 18(4), 55–66. https://doi.org/10.1002/cplx.21441

[40] Chin, R. J., Lai, S. H., Lee, F. W., & Kwong, K. Z. (2023, November 1). Actual shear rate prediction associated with wall slip phenomenon using radial basis function network. The Journal of the Institution of Engineers, Malaysia, 84(1). https://doi.org/10.54552/v84i1.206

[41] Tan, Y., & Zhu, Y. (2010). Fireworks algorithm for optimization. In Y. Tan, Y. Shi & K. C. Tan (Eds.), Advances in swarm intelligence lecture notes in computer science (pp. 355–364). Springer. https://doi.org/10.1007/978-3-642-13495-1_44

[42] Behera, S., Nayak, S. C., & Kumar, A. V. S. P. (2023, September 27). Evaluating the Performance of Metaheuristic Based Artificial Neural Networks for Cryptocurrency Forecasting. Computational Economics. https://doi.org/10.1007/s10614-023-10466-4