Computational Economics
ISSN 2821-0433

Prediction of Bitcoin Cryptocurrency Prices Using Artificial Neural Networks Optimized by Metaheuristic Optimization Algorithms

Aidin Aboutalebi [1], Kambiz Peykarjoo [2*], Ebrahim Rezaei [3], Rahim Khanizad [4]

[1] PhD student in economics, monetary economics, Department of economics, Science and Research Branch, Islamic Azad University, Tehran, Iran, Email: aidin.aboutalebi@iau.ir
[2*] Assistant professor of Economics, Department of Economics, Science and Research Branch, Islamic Azad University, Tehran, Iran, Corresponding Author, Email: k.peykarjou@srbiau.ac.ir
[3] Associate professor of Economics, Department of Economics, Science and Research Branch, Islamic Azad University, Tehran, Iran, Email: Ebrahim.rezaei@gmail.com
[4] Assistant professor of Economics, Department of Economics, Science and Research Branch, Islamic Azad University, Tehran, Iran, Email: rahimkhanizad@gmail.com

Article Info

Received: 11/10/2024
Accepted: 29/12/2024

Pages: 23-55

Keywords:
Cryptocurrency price prediction; Artificial neural network; Bitcoin; Meta-heuristic optimization algorithms

JEL Classification:
F31, C45, D53

ABSTRACT

Cryptocurrencies like Bitcoin, Ethereum, and Litecoin are digital currencies secured by cryptographic technology and based on blockchain, a distributed ledger for transaction verification. Operating independently of banks and governments, cryptocurrencies are widely used for value exchange, investment, and online transactions. Forecasting cryptocurrency prices involves analyzing historical data and market factors to estimate future trends, helping investors and traders make informed decisions. With advancements in computing power, artificial intelligence, particularly artificial neural networks (ANNs), has emerged as a reliable tool for price prediction. This study investigates Bitcoin price prediction over short-term (10-day) and long-term (30-day) intervals using ANNs optimized by six metaheuristic algorithms: Political Optimizer (PO), Heap-Based Optimizer (HBO), Stochastic Paint Optimizer (SPO), Giza Pyramids Construction (GPC), Fire Hawk Optimizer (FHO), and Fox-Inspired Optimization Algorithm (FOX). Performance comparisons of these algorithms revealed that the GPC-optimized ANN consistently outperformed others. In the 10-day interval, the GPC-optimized ANN achieved a determination coefficient of 0.88, demonstrating superior short-term prediction accuracy. Similarly, in the 30-day interval, it achieved a determination coefficient of 0.72, outperforming other models in long-term forecasting as well. These findings highlight the potential of combining ANNs with advanced optimization algorithms to improve cryptocurrency price prediction accuracy, offering valuable insights for traders and investors.

COPYRIGHTS
©2023 by the authors. Published by the Islamic Azad University, West Tehran Branch. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY 4.0) https://creativecommons.org/licenses/by/4.0

Extended Abstract

Purpose

Digital currencies, or cryptocurrencies, are electronic assets that utilize blockchain technology to ensure the security and transparency of transactions. These currencies operate in a decentralized manner and are not controlled by any central authority. Bitcoin, as the first and most well-known cryptocurrency, was created in 2009 by an anonymous individual or group under the name Satoshi Nakamoto. Subsequently, other cryptocurrencies such as Ethereum, Ripple, and Litecoin were developed.

Transactions using cryptocurrencies are often faster and less expensive than traditional banking methods. These digital currencies have become a popular investment option, but their extreme volatility can increase investment risks. In many countries, clear regulations regarding the use of cryptocurrencies have not yet been established, which may pose legal challenges for users. The application of artificial intelligence and machine learning models, particularly artificial neural networks, is crucial for predicting cryptocurrency prices. These digital currencies are accompanied by vast amounts of data, and analyzing them manually can be challenging. AI can process and analyze this extensive data. Artificial neural networks can identify complex, nonlinear patterns in data that may not be detectable by traditional methods. Machine learning models, using advanced algorithms, can enhance the accuracy of cryptocurrency price predictions and assist investors in making better decisions. These models can perform analyses faster than traditional methods, which is highly significant in the volatile cryptocurrency markets.

Methodology

Artificial neural networks consist of numerous processing units called neurons that operate in coordination, similar to human brain neurons, to solve problems. The connections between these neurons determine the network's performance. In these networks, a data structure is designed through programming to act like neurons, and then connections between the neurons are established, followed by the application of a training algorithm to train the network. Artificial neural networks learn similarly to humans through examples.

Artificial neural networks have the ability to adapt to rapid market changes and can identify new patterns. Utilizing AI can help reduce errors caused by emotional decision-making or human biases. These models can simultaneously analyze the impact of various factors, such as economic news, regulatory changes, and social trends. Machine learning models, with access to more data and continuous feedback, can continuously improve their performance. The more precise information and deeper analyses provided by these technologies can serve as powerful tools for investors, minimizing risks. The real-time implementation of these models is possible, which is a significant advantage for active traders.

Based on the aforementioned considerations in this research, the focus has been on predicting Bitcoin prices over short-term (10-day) and long-term (30-day) periods. For this purpose, an artificial neural network optimized by six metaheuristic optimization algorithms has been employed. These algorithms include the Political

Optimizer (PO), Hierarchical Rank-based Optimization (HBO), Stochastic Colors Optimizer (SPO), Giza Pyramid Construction (GPC), Firefly Optimizer (FHO), and Fox Optimizer (FOX). Finally, the results obtained from these algorithms have been compared.

In this study, Bitcoin price data were collected and examined over a 13-year period, from July 13, 2010, to December 6, 2023. This dataset includes 4,895 samples and six variables. The variables include the opening price of Bitcoin, the highest price, the lowest price, trading volume, and market value as inputs, with the final price of Bitcoin considered as the output.

Finding

This research is designed as an applied study, aiming to create a framework for predicting Bitcoin prices over short-term (10-day) and long-term (30-day) periods using the machine learning method known as artificial neural networks, optimized with the six metaheuristic algorithms: PO, HBO, SPO, GPC, FHO, and FOX.

The steps of the work are as follows: Initially, in MATLAB software, the necessary data for modeling is preprocessed, and missing or unavailable values are removed. Then, to reduce the impact of data scaling on the results, data normalization is performed within the range of 0 to 1, as entering raw data can decrease the model's speed and accuracy. Subsequently, 80 percent of the initial Bitcoin price data is divided into training data, while the remaining 20 percent is allocated as test data.

Afterward, using the training dataset, the artificial neural network (ANN) is trained with the mentioned optimization algorithms. In the next stage, the optimized neural network models are tested using the test dataset. Finally, the results and accuracy of various models in predicting Bitcoin prices in both training and testing scenarios are compared. Evaluation criteria include the coefficient of determination (R^2), root mean square error (RMSE), mean square error (MSE), mean absolute error (MAE), relative square error (RSE), and explained variance (EVS). All modeling steps are conducted in MATLAB R2023b.

Conclusion

The results of this research indicate that in the short-term (10-day) period, the neural network optimized with the GPC algorithm achieved a coefficient of determination of 0.88, and in the long-term (30-day) period, this network again performed best compared to other algorithms with a coefficient of determination of 0.7.

فصلنامه اقتصاد محاسباتی
شاپا ۲۸۲۱-۰۴۳۳

پیش‌بینی قیمت رمز ارز بیت‌کوین با استفاده از شبکه عصبی مصنوعی بهینه‌سازی شده توسط الگوریتم‌های بهینه‌سازی فراابتکاری

آیدین ابوطالبی[۱]، کامبیز پیکارجو[۲]✉، ابراهیم رضائی[۳]، رحیم خانی‌زاد[۴]

۱. دانشجوی دکتری، گروه اقتصاد، گرایش اقتصاد پولی، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران، پست الکترونیکی:
aidin.aboutalebi@iau.ir

۲*. استادیار اقتصاد، گروه اقتصاد، ، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران، نویسنده مسئول، پست الکترونیکی:
k.peykarjou@srbiau.ac.ir

۳. دانشیار اقتصاد، گروه اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران، پست الکترونیکی:
Ebrahim.rezaei@gmail.com

۴. استادیار اقتصاد، گروه اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران، پست الکترونیکی:
rahimkhanizad@gmail.com

اطلاعات مقاله

نوع مقاله: مقاله پژوهشی
صفحات ۲۳-۵۵

تاریخ دریافت مقاله: ۱۴۰۳/۰۷/۲۰
تاریخ پذیرش نهایی: ۱۴۰۳/۱۰/۰۹

واژگان کلیدی:
پیش‌بینی قیمت رمز ارز؛ شبکه عصبی مصنوعی؛ بیت‌کوین؛ الگوریتم‌های بهینه‌سازی فراابتکاری

طبقه‌بندی JEL:
F31, C45, D53

چکیده

رمز ارز ها، ارز های دیجیتالی مبتنی بر فناوری بلاک چین هستند که با استفاده از تکنولوژی رمزنگاری، امنیت و کنترل ایجاد واحدهای جدید را تأمین می‌کنند و به صورت مستقل از بانک‌ها و دولت‌ها عمل می‌کنند. بیت کوین، اتریوم و لایت‌کوین از معروف‌ترین رمز ارزها هستند که برای تبادل ارزش و سرمایه‌گذاری استفاده می‌شوند. پیش‌بینی قیمت این ارزها با تحلیل داده‌های تاریخی و عوامل مؤثر بر بازار انجام می‌شود تا به سرمایه‌گذاران در تصمیم‌گیری بهتر کمک نماید. برای موفقیت در بازارهای مالی، تخصص و آگاهی ضروری است؛ بنابراین سرمایه‌گذاران همواره به دنبال روش‌هایی جهت پیش‌بینی دقیق‌تر قیمت رمز ارزها، به‌ویژه بیت کوین هستند. با پیشرفت کامپیوترها و افزایش قدرت پردازش آنها، روش‌های هوش مصنوعی مانند شبکه‌های عصبی مصنوعی به ابزارهایی برای پیش‌بینی دقیق‌تر قیمت رمز ارزها تبدیل شده‌اند. در این پژوهش، به پیش‌بینی قیمت رمز ارز بیت‌کوین در بازه‌های زمانی کوتاه‌مدت (۱۰ روزه) و بلندمدت (۳۰ روزه) با استفاده از شبکه عصبی مصنوعی بهینه‌سازی‌شده توسط شش الگوریتم بهینه‌سازی فراابتکاری به نام‌های الگوریتم بهینه‌ساز سیاسی (PO)، الگوریتم سلسله مراتب رتبه در شرکت (HBO)، الگوریتم بهینه‌ساز رنگ‌های تصادفی (SPO)، الگوریتم ساخت اهرام جیزه (GPC)، الگوریتم بهینه‌ساز شاهین آتشین (FHO) و الگوریتم بهینه‌ساز روباه (FOX) پرداخته شد و نتایج این شش حالت با یکدیگر مقایسه گردید. نتایج نشان داد در بازه زمانی کوتاه‌مدت ۱۰ روزه و بلندمدت ۳۰ روزه، عملکرد پیش‌بینی شبکه عصبی بهینه‌سازی‌شده توسط الگوریتم GPC به‌ترتیب با ضریب تعیین ۰/۸۸ و ۰/۷۲ بهتر از سایر الگوریتم‌ها بوده است.

۱. مقدمه

رمز ارزها، نوعی دارایی دیجیتالی هستند که از فناوری بلاکچین برای امنیت و شفافیت تراکنش‌ها استفاده می‌کنند. این ارزها به صورت غیرمتمرکز عمل کرده و توسط هیچ نهاد مرکزی کنترل نمی‌شوند. بیت‌کوین اولین و معروف‌ترین رمز ارز است. رمز ارزها همچنین به عنوان یک ابزار سرمایه‌گذاری، محبوب شده‌اند اما نوسانات بالایی دارند که می‌تواند ریسک سرمایه‌گذاری را افزایش دهد. استفاده از مدل‌های هوش مصنوعی و یادگیری ماشین، به ویژه شبکه‌های عصبی مصنوعی، در پیش‌بینی قیمت رمز ارزها اهمیت بسیار زیادی دارد. این مدل‌ها قادرند در زمان کوتاه‌تری نسبت به روش‌های سنتی تحلیل انجام دهند، که در بازارهای پرنوسان رمز ارزها، بسیار مهم است (چایتانی و همکاران، ۲۰۲۴)[1]. شبکه‌های عصبی مصنوعی می‌توانند خود را با تغییرات سریع بازار تطبیق دهند و الگوهای جدید را شناسایی کنند. استفاده از هوش مصنوعی می‌تواند خطاهای ناشی از تصمیمات احساسی یا سوگیری انسانی را کاهش دهد. این مدل‌ها می‌توانند تاثیر متغیرهای مختلف مانند اخبار اقتصادی، تغییرات قانونی و روندهای اجتماعی را همزمان بررسی کنند (چاجر و همکاران،۲۰۲۲)[2]. با توجه به موارد ذکر شده در این پژوهش به پیش‌بینی قیمت رمز ارز بیت کوین در بازه‌های زمانی کوتاه‌مدت ۱۰روزه و بلندمدت ۳۰ روزه با استفاده از شبکه عصبی مصنوعی بهینه‌سازی شده توسط شش الگوریتم بهینه‌سازی فراابتکاری به نام‌های الگوریتم PO، HBO، SPO، GPC، FHO و FOX پرداخته شده و نتایج آن‌ها با یکدیگر مقایسه شده است.

۲. ادبیات موضوع

به منظور پیش‌بینی قیمت رمز ارزها با استفاده از مدل‌های هوش مصنوعی و یادگیری ماشین، مطالعات داخلی و خارجی زیادی انجام شده است که در ادامه به برخی از آن‌ها اشاره شده است.

صیادی نژاد و همکاران (۱۴۰۲)، در مقاله خود، مدل هیبریدی CEEMD-DL(LSTM) با ترکیب تکنیک‌های یادگیری عمیق و روش تجزیه مد تجربی یکپارچه کامل[3] برای پیش‌بینی بازده قیمت بیت‌کوین، که به عنوان یکی از پرطرفدارترین ارزهای دیجیتال شناخته می‌شود، به کار گرفته‌اند. آنها داده‌های روزانه قیمت بیت‌کوین در بازه زمانی ۲۰۱۳/۰۱/۰۱ تا ۲۰۲۲/۰۵/۲۸ راجمع‌آوری نموده و نتایج به‌دست‌آمده را با سایر مدل‌های رقابتی بر اساس معیارهای عملکرد مقایسه کرده اند. یافته‌ها نشان‌دهنده این است که استفاده از مدل CEEMD-DL(LSTM) باعث بهبود کارایی و دقت پیش‌بینی بازدهی

[1] Choithani et al., 2024
[2] Chhajer et al., 2022
[3] CEEMD

بیت‌کوین شده است. بنابراین، پیشنهاد می‌شود که این مدل برای پیش‌بینی در این زمینه مورد استفاده قرار گیرد (صیادی نژاد و همکاران،۱۴۰۲)

دووتا و همکاران[1] (۲۰۲۰)، یک ساختار نوین با به کارگیری مجموعه‌ای از روش‌های پیشرفته یادگیری ماشین برای پیش‌بینی قیمت روزانه بیت‌کوین را ارائه کردند. یافته‌های آن‌ها نشان می‌دهد که مدل‌های شبکه‌های عصبی بازگشتی، عملکردی برتر نسبت به مدل‌های سنتی یادگیری ماشین دارند و معماری شبکه عصبی واحد بازگشتی در تحلیل داده‌ها، قابلیت بیشتری نسبت به شبکه عصبی از نوع حافظه کوتاه مدت بلند نشان می‌دهد.

سوی و همکاران[2] (۲۰۲۴)، در مطالعه خود بر بهبود پیش‌بینی قیمت سهام برای سرمایه‌گذاران خُرد با استفاده از تکنیک‌های پیشرفته یادگیری ماشین بر اساس داده‌های بازار بورس پرداختند. آن‌ها از یک روش جامع شامل پیش‌پردازش داده‌ها برای مدیریت مقادیر گمشده و داده‌های پرت، مهندسی ویژگی، اعتبارسنجی متقابل و تنظیم پارامترها استفاده کردند. تکنیک‌های به‌کار رفته شامل شبکه‌های عصبی عمیق Keras، LightGBM، LSTM، GRU و رگرسیون خطی بوده است. مدل ترکیبی پیشنهادی آن‌ها که مدل‌های سری زمانی و یادگیری عمیق را ترکیب می‌کند، عملکرد بهتری نسبت به مدل‌های فردی نشان می‌دهد. این ادغام روش‌ها منجر به بهبود قابل توجهی در دقت پیش‌بینی می‌شود و راه‌حلی قوی برای سرمایه‌گذاران خُرد فراهم می‌کند.

ایبلدیز و اسکندراوغلو[3] (۲۰۲۴)، در پژوهش خود به مقایسه عملکرد الگوریتم‌های مختلف در پیش‌بینی جهت حرکت شاخص‌های بازار سهام کشورهای توسعه‌یافته و تعیین بهترین الگوریتم پرداختند. شاخص‌هایی همچون NYSE 100 (آمریکا)، NIKKEI 225 (ژاپن)، FTSE 100 (انگلستان)، CAC 40 (فرانسه)، DAX 30 (آلمان)، FTSE MIB (ایتالیا) و TSX (کانادا) با الگوریتم‌هایی مانند درخت تصمیم، جنگل تصادفی، نزدیک‌ترین همسایه، بیز ساده، رگرسیون لجستیک، ماشین بردار پشتیبان و شبکه عصبی مصنوعی پیش‌بینی شدند. نتایج نشان داد که شبکه عصبی مصنوعی برای شاخص‌های NYSE 100، FTSE 100، DAX 30 و FTSE MIB و رگرسیون لجستیک برای شاخص‌های NIKKEI 225، CAC 40 و TSX بهترین عملکرد را داشتند. همچنین، شبکه عصبی مصنوعی به عنوان بهترین الگوریتم با دقت پیش‌بینی بالاتر از ۷۰ درصد برای تمامی شاخص‌ها شناخته شد.

۳. روش تحقیق

در این بخش ابتدا به معرفی مجموعه داده‌های مورد استفاده در این پژوهش پرداخته می‌شود، سپس

[1] Dutta et al.,2020
[2] Sui et al.,2024
[3] Ayyildiz & Iskenderoglu 2024

روش یادگیری ماشین شبکه عصبی مصنوعی و شش الگوریتم بهینه‌سازی فراابتکاری به منظور بهینه‌سازی شبکه‌های عصبی مصنوعی مورد استفاده در این مطالعه شرح داده می‌شود.

۳-۱. مجموعه داده مورد استفاده و آماده سازی داده‌ها

مجموعه داده‌های قیمت بیت کوین مورد استفاده در این تحقیق مربوط به یک دوره‌ی ۱۳ ساله از ۱۳ جولای ۲۰۱۰ تا ۶ دسامبر ۲۰۲۳ جمع آوری و مورد استفاده قرار گرفته است. این مجموعه داده شامل ۴۸۹۵ نمونه و ۶ متغیر می‌باشند. متغیرهای قیمت رمز ارز بیت کوین در هنگام باز شدن[1]، قیمت بالایی رمز ارز[2]، قیمت پایینی رمز ارز[3]، حجم معاملات[4] و ارزش بازار ارز[5] به عنوان متغیرهای ورودی و متغیر قیمت نهایی رمز ارز بیت کوین[6] به عنوان متغیر خروجی در نظر گرفته شده است.

جدول (۱): مقادیر آمار توصیفی ویژگی‌های مجموعه داده قیمت رمز ارز بیت کوین

variables	Open	High	Low	Volume	Market Cap	Close
count	۴۸۹۵	۴۸۹۵	۴۸۹۵	۴۸۹۵	۴۸۹۵	۴۸۹۵
mean	۹۹۲۸/۴۲	۱۰۱۶۳/۱۸	۹۶۷۶/۳۴	۱۱۳۵۴۵۹۸۴۸۵/۶۱	۱۸۵۳۴۳۸۷۸۶۲۵/۲۷	۹۹۳۶/۴۴
std	۱۴۸۶۰/۹۹	۱۵۲۱۹/۷۹	۱۴۴۶۳/۱۵	۱۷۶۱۸۸۹۰۶۸۰/۴۷	۲۸۲۰۰۸۷۴۳۶۹۴/۵۷	۱۴۸۶۵/۶۸
min	۰/۰۶	۰/۰۶	۰/۰۵	۰/۰۰	۱۹۰۲۶۰/۰۰	۰/۰۶
25%	۲۲۰/۰۱	۲۲۶/۲۶	۲۱۴/۸۰	۱۵۰۹۸۵/۵۰	۳۰۵۰۶۲۹۸۸۳/۵۰	۲۱۹/۹۸
50%	۱۱۷۲/۷۱	۱۱۹۱/۱۰	۱۱۵۶/۴۴	۳۱۵۱۰۸۰۰۰/۰۰	۱۹۱۰۱۴۲۷۵۱۸/۰۰	۱۱۷۵/۹۵
75%	۱۱۵۹۶/۴۸	۱۱۸۹۸/۶۵	۱۱۳۰۰/۳۰	۱۹۴۷۳۸۴۷۹۲۲/۵۰	۲۱۲۰۸۵۹۲۶۷۸۰/۵۰	۱۱۶۰۰/۷۹
max	۶۷۵۴۹/۷۴	۶۸۷۸۹/۶۳	۶۶۳۸۲/۰۶	۳۵۰۹۶۷۹۴۱۴۷۹/۰۰	۱۲۷۴۸۳۱۴۹۰۸۵۱/۰۰	۶۷۵۶۶/۸۳

منبع: یافته های تحقیق

۳-۲. شبکه عصبی مصنوعی

شبکه‌های عصبی مصنوعی از شمار بسیار زیادی عناصر پردازش کننده به‌هم پیوسته به نام نورون تشکیل شده‌اند که برای حل یک مسئله همانند نورون‌های مغز انسان به صورت هماهنگ با هم عمل می‌کنند. ارتباط بین این نورون‌ها عملکرد شبکه را تعیین می‌کند. در این شبکه‌ها به کمک دانش برنامه‌نویسی، ساختار داده‌ای طراحی می‌شود که می‌تواند همانند نورون عمل کند، سپس با ایجاد

[1] Open
[2] High
[3] Low
[4] Volume
[5] Market Cap
[6] Close

شبکه‌ای بین گره‌ها و اعمال یک الگوریتم آموزشی به آن، شبکه را آموزش می‌دهند. شبکه‌های عصبی مصنوعی همانند انسان با استفاده از مثال‌ها آموزش می‌بینند (اسمیت،۱۹۹۳)[1]. در حالت کلی در شبکه‌های عصبی، سه لایه‌ی نرونی وجود دارد. لایه‌ی ورودی که وظیفه دریافت اطلاعات خامی که به شبکه تغذیه شده است را بر عهده دارد. لایه‌های پنهان که عملکرد این لایه به‌وسیله‌ی ورودی‌ها و وزن ارتباط بین آن‌ها و لایه‌های پنهان تعیین می‌شود. وزن‌های بین واحدهای ورودی و پنهان تعیین می‌کند که چه زمان یک واحد پنهان باید فعال شود. لایه خروجی که عملکرد واحد خروجی بسته به فعالیت واحد پنهان و وزن ارتباط بین واحد پنهان و خروجی می‌باشد (هورنیک و همکاران، ۱۹۸۹)[2].

در یک شبکه عصبی مصنوعی، نورون mام اطلاعات ورودی خود را از طریق گره‌های ورودی iI دریافت می‌کند. هر یک از این گره‌های ورودی، قبل از اینکه وارد هسته اصلی نورون شود، وزن‌دار می‌شوند. یعنی مقدار هر ورودی در Wim ضرب می‌شود. سپس این مقادیر در بخش اول پردازش‌گر با هم جمع شده و مجموع کل ورودی به نورون تعیین می شود . در برخی موارد، یک مقدار ثابت در هر نورون بنام وزن اریب یا بایاس به مقدار کل ورودی افزوده می‌شود. مق–دار این ورودی یک و وزن آن b است. با در نظر گرفتن این وزن بایاس، مقدار کل ورودی به نورون از رابطه زیر محاسبه می‌شود (منهاج، ۱۳۸۱).

$$U_m = \sum_{i=1}^{n} I_i W_{i,m} + b_m \qquad (۱)$$

در مرحله بعد، یک تابع موسوم به تابع فعال سازی که معمولا تابعی غیرخطی است، روی این مقدار حاصل جمع عمل کرده و مقدار خروجی نورون از رابطه زیر تعیین می‌گردد (منهاج، ۱۳۸۱).

$$O_m = f(\sum_{i=1}^{n} I_i W_{i,m} + b_m) \qquad (۲)$$

هدف از به کار بردن این تابع، محدود کردن خروجی در یک باز–د مشخص اس–ت. ب–ه عبارت دیگر با وجود این تابع، خروجی یک نورون در مقابل ورودی‌ها ی بسیار کوچک یا بسیار بزرگ از حدود معینی تجاوز نمی‌کند (منهاج، ۱۳۸۱).

۳-۳. الگوریتم بهینه‌ساز سیاسی

الگوریتم بهینه ساز سیاسی[3] PO در سال ۲۰۲۰ توسط عسکری و همکاران ابداع شد. الگوریتم PO به عنوان یک دنباله از پنج فاز شامل تشکیل حزب و تخصیص حوزه انتخابیه، کمپین انتخاباتی، تغییر حزب، انتخابات بین حزبی و امور پارلمانی ساختار یافته است (عسکری و همکاران،۱۳۹۸) . فاز تشکیل حزب و تخصیص حوزه انتخابیه فقط یک بار اجرا می‌شود و بقیه چهار فاز در یک حلقه اجرا می‌شوند. جمعیت P به n حزب سیاسی تقسیم می‌شود (رابطه (۳)). هر حزب P_i شامل n نامزد/عضو است (رابطه (۴)). هر

[1] Smith, 1993
[2] Hornik et al., 1989
[3] Political Optimizer (PO)

عضو j از P_i^j به عنوان یک راه حل بالقوه در نظر گرفته می‌شود که یک بردار d –بعدی است (رابطه (۵)). در رابطه مقدار d تعداد متغیرهای ورودی مسئله در حال حل است و $P_{i,k}^j$بعد k ام از P_i^j را نشان می‌دهد.

$$P = \{P_1, P_2, P_3, \dots, P_n\} \quad (۳)$$

$$P_i = \{P_i^1, P_i^2, P_i^3, \dots, P_i^n\} \quad (۴)$$

$$P_i^j = \left[P_{i,1}^j, P_{i,2}^j, P_{i,3}^j, \dots, P_{i,d}^j\right]^T \quad (۵)$$

علاوه بر نقش عضو یک حزب، یک راه حل بالقوه نیز به عنوان یک نامزد انتخابات عمل می‌کند. فرض می‌شود که n حوزه انتخابیه وجود دارد (رابطه (۶)) و عضو j ام هر حزب از حوزه انتخابیه jام Cj رقابت می‌کند (رابطه (۷)).

$$C = \{C_1, C_2, C_3, \dots, C_n\} \quad (۶)$$

$$C_i = \{P_1^j, P_2^j, P_3^j, \dots, P_n^j\} \quad (۷)$$

متناسب‌ترین عضو یک حزب به عنوان رهبر حزب اعلام می‌شود که بلافاصله پس از انتخابات عمومی (انتخابات بین حزبی) تصمیم گرفته می‌شود (رابطه (۸)) که در آن P_i^* رهبر حزب i را نشان می‌دهد و $f\left(P_i^j\right)$ برازندگی P_i^j را محاسبه می‌کند. مجموعه تمام رهبران حزب با P^* نمایش داده می‌شود (رابطه (۹)). پس از انتخابات، برندگان از تمام حوزه‌های انتخابیه به عنوان نمایندگان پارلمان می‌شوند. C^* مجموعه تمام نمایندگان پارلمان را نشان می‌دهد و C_J^* برنده حوزه انتخابیه j را نشان می‌دهد (رابطه (۱۰)).

$$q = \underset{1\le j\le n}{argminf}\left(P_i^j\right), \forall i \in \{1, \dots, n\}\ P_i^* = P_i^q \quad (۸)$$

$$P^* = \{P_1^*, P_2^*, P_3^*, \dots, P_n^*\} \quad (۹)$$

$$C^* = \{C_1^*, C_2^*, C_3^*, \dots, C_n^*\} \quad (۱۰)$$

۳-۳-۱. کمپین انتخاباتی (کاوش و استخراج)

این فاز به نامزدها کمک می‌کند تا عملکرد خود در انتخابات را بهبود بخشند که از روابط (۱۱) و (۱۲) برای به‌روزرسانی موقعیت یک نامزد استفاده می‌کند. بسته به رابطه برازندگی فعلی $f\left(P_i^j(t)\right)$ یک نامزد با برازندگی قبلی خود $f\left(P_i^j(t-1)\right)$، تصمیم گرفته می‌شود که آیا موقعیت را با استفاده از رابطه (۱۱) یا رابطه (۱۲) به‌روز کنیم. اگر برازندگی بهبود یابد، سپس از رابطه (۱۱) برای به‌روزرسانی موقعیت استفاده می‌شود، با این حال، اگر برازندگی بدتر شود، سپس از رابطه (۱۲) استفاده می‌شود.

$$p_{i,k}^{j}(t+1)=\begin{cases} m^{*}+r\left(m^{*}-p_{i,k}^{j}(t)\right), & if\ \ p_{i.k}^{j}(t-1)\le p_{i,k}^{j}(t)\le m^{*}\ or\ \ p_{i.k}^{j}(t-1)\ge p_{i.k}^{j}(t)\ge m^{*} \\ m^{*}+(2r-1)\left|m^{*}-p_{i,k}^{j}(t)\right|, & if\ \ p_{i.k}^{j}(t-1)\le m^{*}\le p_{i.k}^{j}(t)\ or\ \ p_{i.k}^{j}(t-1)\ge m^{*}\ge p_{i,k}^{j}(t) \\ m^{*}+(2r-1)\left|m^{*}-p_{i,k}^{j}(t)\right|, & if\ \ m^{*}\le\ p_{i.k}^{j}(t-1)\le p_{i.k}^{j}(t)\ or\ \ m^{*}\ge p_{i.k}^{j}(t-1)\ge p_{i,k}^{j}(t) \end{cases} \quad (۱۱)$$

$$p_{i,k}^{j}(t+1)=\begin{cases} m^{*}+(2r-1)\left|m^{*}-p_{i,k}^{j}(t)\right|, & if\ \ p_{i.k}^{j}(t-1)\le p_{i,k}^{j}(t)\le m^{*}\ or\ \ p_{i.k}^{j}(t-1)\ge p_{i.k}^{j}(t)\ge m^{*} \\ p_{i,k}^{j}(t-1)+r\left(p_{i,k}^{j}(t)-p_{i,k}^{j}(t-1)\right), & if\ \ p_{i.k}^{j}(t-1)\le m^{*}\le p_{i.k}^{j}(t)\ or\ \ p_{i.k}^{j}(t-1)\ge m^{*}\ge p_{i,k}^{j}(t) \\ m^{*}+(2r-1)\left|m^{*}-p_{i,k}^{j}(t-1)\right|, & if\ \ m^{*}\le\ p_{i.k}^{j}(t-1)\le p_{i.k}^{j}(t)\ or\ \ m^{*}\ge p_{i.k}^{j}(t-1)\ge p_{i,k}^{j}(t) \end{cases} \quad (۱۲)$$

۳-۳-۲. تغییر حزب (تعادل بین کاوش و استخراج)

یک پارامتر تطبیقی λ به نام نرخ تغییر حزب تعریف شده است که از λ_{max} شروع می‌شود و به طور خطی به ۰ در طول تکرارها کاهش می‌یابد. هر عضو P_i^j با احتمال λ انتخاب می‌شود و به حزب تصادفی P_r تغییر می‌کند، که در آن با کمترین عضو مناسب P_r^q از آن حزب P_r جابجا/مبادله می‌شود. محاسبه شاخص q از عضو کمترین مناسب P_r در رابطه (۱۳) بیان شده است.

$$q=\underset{1\le j\le n}{argminf}\left(P_i^j\right) \quad (۱۳)$$

۳-۳-۳. انتخابات (ارزیابی برازندگی)

انتخابات با ارزیابی برازندگی تمام نامزدهای رقابت کننده در یک حوزه انتخابیه و اعلام برنده انجام می‌شود (رابطه (۱۴)). رابطه (۱۴) برنده حوزه انتخابیه j ام (Cj) را نشان می‌دهد. همانطور که قبلاً اشاره شد، رهبران حزب نیز پس از انتخابات با استفاده از رابطه (۸) به‌روز می‌شوند.

$$q=\underset{1\le i\le n}{argminf}\left(P_i^j\right) \quad (۱۴)$$
$$\mathrm{c}_j^{*}=P_q^j$$

۳-۳-٤. امور پارلمانی (استخراج و همگرایی)

پس از انتخابات بین حزبی، دولت تشکیل می‌شود. رهبران حزب و برندگان حوزه انتخابیه/نمایندگان پارلمان با استفاده از روابط (۸) و (۱۴) تصمیم گرفته می‌شوند. هر نماینده پارلمان c_j^{*} (برنده حوزه انتخابیه j ام) موقعیت خود را با مراجعه به یک نماینده پارلمان تصادفی c_r^{*} به‌روز می‌کند و اگر این باعث هرگونه بهبود در برازندگی c_j^{*} شود، وضعیت و برازندگی c_j^{*} به‌روز می‌شود.

۳-٤. الگوریتم بهینه‌ساز سلسله مراتب رتبه در شرکت

در یک سازمان، گروهی از افراد که برای یک هدف مشترک کار می‌کنند، ممکن است موفق نشوند مگر اینکه خود را در یک سلسله‌مراتب به نام سلسله‌مراتب رتبه در شرکت[1] (CRH) سازماندهی کنند. الگوریتم پیشنهادی به نام بهینه‌ساز مبتنی بر هرم[2] (HBO) نامیده می‌شود زیرا از ساختار داده هرم برای نقشه‌برداری مفهوم CRH استفاده می‌کند. مدل ریاضی الگوریتم HBO بر سه اصل استوار است: تعامل بین زیردستان و رئیس فوری آن‌ها، تعامل بین همکاران، و خودکمکی کارکنان (عسکری و همکاران،۲۰۲۰)[3].

۳-٤-۱. مدل‌سازی ریاضی تعامل با رئیس فوری

با فرض اینکه هر گره والد یک رئیس فوری به فرزندانش است، این رفتار می‌تواند با به‌روزرسانی موقعیت هر عامل جستجو $\vec{X}_i$ با ارجاع به گره والد B با استفاده از رابطه زیر مدل شود:

$$X_i^k(t+1) = B^k + \gamma\lambda^k\left|B^k - X_i^k(t)\right| \quad (۱۵)$$

که در آن t نشان‌دهنده تکرار فعلی است، k در بالانویس نشان‌دهنده کامپوننت k ام یک بردار است، و | | مقدار مطلق را محاسبه می‌کند. λ^k کامپوننت k ام بردار λ است که به صورت تصادفی به شرح زیر تولید می‌شود:

$$\lambda^k = 2r - 1 \quad (۱۶)$$

که در آن r یک عدد تصادفی است که بر اساس توزیع یکنواخت از محدوده [0,1] تولید می‌شود. در رابطه (۱۵)، γ یک پارامتر با دقت طراحی شده است که به شرح زیر محاسبه می‌شود:

$$\gamma = \left|2 - \frac{\left(t - mod\frac{T}{C}\right)}{\frac{T}{4C}}\right| \quad (۱۷)$$

که در آن t نشان‌دهنده تکرار فعلی است، T نمایانگر تعداد کل تکرارها است، و C یک پارامتر تعریف شده توسط کاربر است و در زیر توضیح داده شده است. با گذر تکرارها، γ به طور خطی از ۲ به ۰ کاهش می‌یابد و پس از رسیدن به ۰، شروع به افزایش دوباره به ۲ با تکرارها می‌کند. با این حال، پارامتر C است که تعیین می‌کند چند دور γ در T تکرار کامل خواهد شد.

[1] Corporate Rank Hierarchy
[2] Heap-based optimizer (HBO)
[3] Askari et al., 2020

۳-٤-۲. مدل‌سازی ریاضی تعامل بین همکاران

مقامات با یک رتبه به عنوان همکاران در نظر گرفته شده‌اند. آن‌ها با یکدیگر برای انجام وظایف رسمی تعامل می‌کنند. در هرم، ما فرض می‌کنیم که گره‌ها در یک سطح همکار هستند و هر عامل جستجو $\vec{X}_i$ موقعیت خود را با ارجاع به همکار تصادفی انتخاب شده $\vec{S}_r$ بر اساس رابطه زیر به‌روز می‌کند که در آن f نشان‌دهنده تابع هدف است و برازندگی عامل جستجو را محاسبه می‌کند.

$$X_i^k(t+1) = \begin{cases} S_r^k + \gamma\lambda^k\left|S_r^k - X_i^k(t)\right|, & f(\vec{S}_r) < f(\vec{X}_i(t)) \\ X_i^k + \gamma\lambda^k\left|S_r^k - X_i^k(t)\right|, & f(\vec{S}_r) \geq f(\vec{X}_i(t)) \end{cases} \quad (۱۸)$$

۳-٤-۳. مدل‌سازی خودکمکی یک کارمند

این فاز مفهوم خودکمکی یک کارمند را نقشه می‌کشد. نقشه‌برداری این فاز بسیار ساده نگه داشته شده است. این رفتار با حفظ موقعیت قبلی کارمند در تکرار بعدی، به شرح زیر مدل می‌شود:

$$X_i^k(t+1) = X_i^k(t) \quad (۱۹)$$

۳-٤-٤. ادغام تمام مراحل قبل

در این بخش ما بحث می‌کنیم که چگونه می‌توان معادلات به‌روزرسانی موقعیت، که در زیربخش‌های قبلی مدل شده‌اند، را در یک معادله ادغام کرد. یک چالش اصلی تعیین احتمالات انتخاب برای همه سه معادله است زیرا احتمالات انتخاب آن‌ها نقش قابل توجهی در تعادل کاوش و استخراج بازی می‌کند. یک چرخ رولت تعادل این احتمالات را طراحی کرده است، که به سه نسبت p_1، p_2، و p_3 تقسیم شده است. انتخاب نسبت p_1 به یک عامل جستجو اجازه می‌دهد تا موقعیت خود را با استفاده از رابطه (۱۹) به‌روز کند. حد p_1 به شرح زیر محاسبه می‌شود:

$$P_1 = 1 - \frac{t}{T} \quad (۲۰)$$

که در آن t نشان‌دهنده تکرار فعلی است و T نشان‌دهنده حداکثر تعداد تکرارها است. انتخاب نسبت p2 به یک عامل جستجو اجازه می‌دهد تا موقعیت خود را با استفاده از رابطه (۱۵) به‌روز کند. حد p2 به شرح زیر محاسبه می‌شود:

$$P_2 = P_1 + \frac{1-P_1}{2} \quad (۲۱)$$

در نهایت، انتخاب p3 نشان‌دهنده به‌روزرسانی موقعیت با استفاده از رابطه (۱۸) است و حد p3 به شرح زیر محاسبه می‌شود:

$$P_2 = P_1 + \frac{1-P_1}{2} \quad (۲۲)$$

مکانیزم به‌روزرسانی موقعیت عمومی HBO در رابطه زیر ارائه شده است که در آن p یک عدد تصادفی در محدوده [0,1] است.

$$X_i^k(t+1) = \begin{cases} X_i^k(t) & p \le p_1 \\ B^k + \gamma\lambda^k\left|B^k - X_i^k(t)\right|, & p > p_1 \; and \; p \le p_2 \\ S_r^k + \gamma\lambda^k\left|S_r^k - X_i^k(t)\right|, & p > p_2 \; and \; p \le p_3 \;\; and \;\; f(\vec{S}_r) < f(\vec{X}_i(t)) \\ X_i^k + \gamma\lambda^k\left|S_r^k - X_i^k(t)\right|, & p > p_2 \; and \; p \le p_3 \; and \;\; f(\vec{S}_r) \ge f(\vec{X}_i(t)) \end{cases} \quad (۲۳)$$

۳-۵. الگوریتم بهینه‌ساز رنگ‌های تصادفی

الگوریتم بهینه ساز رنگ‌های تصادفی (SPO) الهام‌گرفته از هنر است. SPO یک بهینه‌ساز مبتنی بر جمعیت است که الهام‌گرفته از هنر نقاشی است و زیبایی رنگ‌ها نقش اصلی را در این الگوریتم ایفا می‌کند. SPO به عنوان یک الگوریتم بهینه‌سازی، فضای جستجو را به عنوان یک بوم نقاشی شبیه‌سازی می‌کند و ترکیب رنگ متفاوتی را برای پیدا کردن بهترین رنگ اعمال می‌کند (کاوه و همکاران،۱۴۰۰)

۳-۵-۱. فاز ۱: مقداردهی اولیه

در یک جستجوی nc–بعدی، رنگ‌های اولیه تمام نقاشی‌ها به صورت تصادفی تعیین می‌شوند.

$$C_{i,0} = C(Cmin_{max},\ i = 1,2,3,\dots,nc)_{min} \quad (۲۴)$$

که در آن $C_{i,0}$رنگ اولیه نقاشی i است. C_{min} و C_{max} حد پایین و بالای متغیر طراحی i هستند، $rand$ یک عدد تصادفی در بازه [۰, ۱] است و nc تعداد رنگ‌ها یا متغیرها است. باید توجه داشت که تمام رنگ‌ها با یکدیگر یک نقاشی را ایجاد می‌کنند که یک طراحی یا راه‌حل مسائل بهینه‌سازی است. سپس، ارزیابی تابع هدف برای هر نقاشی به دست می‌آید. به این ترتیب زیبایی هر نقاشی مشخص می‌شود.

۳-۵-۲. فاز ۲: ارزیابی، مرتب‌سازی و خوشه‌بندی

نقاشی‌ها به عنوان نتیجه مسئله بر اساس تابع هدف مربوطه به ترتیب افزایشی مرتب می‌شوند. در نهایت، آن‌ها به سه گروه برابر یعنی اصلی (بهترین)، فرعی(خوب) و سومی(بدترین) خوشه بندی می‌شوند، به این ترتیب، نیازی به افزودن پارامترها در الگوریتم برای خوشه‌بندی نیست.

۳-۵-۳. فاز ۳: استفاده از تکنیک‌های ترکیب

در این گام، نقاشی‌های جدید با استفاده از چهار تکنیک ترکیب مختلف ارائه شده، ایجاد می‌شوند. تکنیک‌های اصلی ترکیب رنگ بر اساس چرخه رنگ تعریف شده‌اند. تمام این تکنیک‌ها برای ایجاد رنگ‌های جدید C_{new} استفاده می‌شوند. تکنیک ترکیب مشابه طرح‌های رنگ مشابه رنگ‌هایی را که

نزدیک به یکدیگر (همسایگی) در چرخه رنگ هستند، از جمله بنفش – سبز، بنفش و زرد– نارنجی که کنار هم قرار دارند، ترکیب می‌کنند (رابطه (۲۵)).

$$C_{new,1} = C_i + rand.(C_{i+1} - C_{i-1}) \quad (۲۵)$$

تکنیک ترکیب مکمل رنگ‌هایی که در مقابل یکدیگر در چرخه قرار دارند به عنوان رنگ‌های مکمل شناخته می‌شوند را ترکیب می‌کند (رابطه (۲۶)).

$$C_{new,2} = C_i + rand.(C_{Pi} - C_{Ti}) \quad (۲۶)$$

تکنیک ترکیب سه‌گانه، طرح رنگ سه‌گانه هر سه رنگی است که در چرخه رنگ به یک فاصله از یکدیگر هستند. به عنوان مثال، قرمز، زرد و آبی می‌توانند براساس این تکنیک ترکیب شوند (رابطه (۲۷)).

$$C_{new,3} = C_i + rand.\left(\frac{C_{Pi}+C_{Si}+C_{Ti}}{3}\right) \quad (۲۷)$$

تکنیک ترکیب چهارگانه، در نهایت، آخرین تکنیک ترکیبی که توسط نقاشان استفاده می‌شود، چهارگانه است. چهار رنگ که در دو جفت مکمل مستطیلی یا چهارگانه گروه‌بندی شده‌اند، استفاده می‌شوند (رابطه (۲۸)).

$$C_{new,4} = C_i + (rand_1.C_{Pi} + rand_2.C_{Ti} + rand_3.C_{Si} + rand_4.C_{rand})/4 \quad (۲۸)$$

۳-۵-٤. فاز ٤: ارزیابی و به‌روزرسانی

نقاشی‌های جدید ارزیابی می‌شوند و اگر شاخص زیبایی جدید آن‌ها از قبلی بهتر باشد، به جای قدیمی جایگزین می‌شود.

۳-۵-۵. فاز ۵: بررسی شرایط پایان الگوریتم

پس از مجموعه‌ای از تکرارها، چرخه بهینه‌سازی خاتمه می‌یابد. اگر معیار برآورده نشود، یک فرآیند جدید برای فاز۲ برنامه‌ریزی می‌شود؛ در غیر این صورت، فرآیند متوقف می‌شود و بهترین راه‌حل گزارش می‌شود.

۳-٦. الگوریتم ساخت اهرام جیزه

الگوریتم ساخت اهرام جیزه[1] (GPC) با الهام از گذشته باستان دارای ویژگی‌های یک الگوریتم فراابتکاری

[1] Giza Pyramids Construction (GPC)

خوب برای مقابله با بسیاری از مسائل است.الگوریتم پیشنهادی با حرکات کارگران و هل دادن بلوک‌های سنگی روی سطح شیب دار کنترل می‌شود (حریفی و همکاران،۱۳۹۹) . اهرام با استفاده از رمپ مستقیم ساخته شده‌اند. فرض بر این است که فقط از یک رمپ استفاده می‌شود. در الگوریتم زاویه‌ای که رمپ با افق ایجاد می‌کند کمتر از ۱۵ درجه است و می‌تواند متغیر باشد. (باستان شناسان بر این باورند که زاویه بین ۸ تا ۱۲ درجه بوده است). راه حل‌ها از حاصل موقعیت کارگر و بلوک سنگ به دست می‌آیند. چون کارگر در واقع سنگ را هل می‌دهد. اصطکاک در جابجایی بلوک سنگ موثر است اما برای کارگران در نظر گرفته نمی‌شود.

f_k نیروی اصطکاک جنبشی است و f_k را می توان از رابطه زیر به دست آورد.

$$f_k = \mu_k mg\cos\theta \qquad (۲۹)$$

که m جرم بلوک سنگ، g گرانش زمین، زاویه‌ای است که سطح شیب دار با افق ایجاد می کند، و k ضریب اصطکاک جنبشی است. از آنجایی که ما روی محور x قرار داریم، طبق قانون دوم نیوتن یعنی $\sum \vec{F} = m\vec{a}$ داریم:

$$-mg\sin\theta - f_k = ma \qquad (۳۰)$$

که a شتاب است. با قرار دادن رابطه (۲۹) در رابطه (۳۰)، شتاب بلوک سنگ به سمت بالا در سطح شیب‌دار به دست می‌آید.

$$-mg\sin\theta - f_k = ma \qquad (۳۱)$$

بنابراین، در اینجا به یک معادله حرکت مستقل از زمان تحت شتاب ثابت نیاز داریم که با استفاده از معادله زیر می‌توان جابجایی یک بلوک سنگ را روی سطح شیبدار محاسبه کرد.

$$d = \frac{v_0^2}{2g(\sin\theta + \mu_k\cos\theta)} \qquad (۳۲)$$

که d مقدار جابجایی است. همانطور که قبلا ذکر شد g گرانش زمین است که مقدار آن ۹/۸ است. θ زاویه ای است که رمپ با افق ایجاد می کند. v_0 سرعت اولیه بلوک سنگ است و در الگوریتم با یک عدد تصادفی توزیع شده یکنواخت در هر تکرار تعیین می شود.

$$v_0 = rand(0,1) \qquad (۳۳)$$

در واقع، $rand(0,1)$ یک عدد تصادفی بین ۰ و ۱ است، یعنی $0 < v_0 = rand(0,1) < 1$ همچنین μ_k ضریب جنبشی اصطکاک بین بلوک سنگ و سطح شیب دار است و در الگوریتم با عدد تصادفی توزیع شده یکنواخت تعیین می‌شود.

$$\mu_k = rand[\mu k_max_{k_min}] \quad (۳۴)$$

در الگوریتم، حداقل μ_k و ماکزیمم μ_k از قبل تعیین شده‌اند، سپس یک عدد تصادفی بین این دو مقدار در هر تکرار در نظر گرفته می‌شود. به عبارت دیگر، $\mu_{k_min} \le \mu_k \le \mu_{k_max}$ است. دلیل تصادفی بودن میزان اصطکاک، صیقلی نبودن سطح رمپ است و به دلیل ناهمواری احتمالی در برخی قسمت‌ها، ممکن است اصطکاک کم یا زیاد شود. رابطه (۳۲) مقدار جابجایی بلوک سنگ را نسبت به موقعیت قبلی آن تعیین می‌کند. این رابطه با کمی تغییر برای تعیین موقعیت جدید کارگر استفاده می‌شود. برای کارگر اصطکاک در نظر گرفته نمی‌شود. در این رابطه x مقدار حرکت کارگر است.

$$x = \frac{v_0^2}{2g \sin\theta} \quad (۳۵)$$

پس از محاسبه تغییرات جابجایی بلوک سنگ و حرکت کارگر از طریق روابط (۳۲) و (۳۵)، از حاصل این دو معادله می توان موقعیت جدیدی به دست آورد. این موقعیت جدید یک راه حل جدید است.

$$\vec{p} = (\vec{p}_i + d)x\vec{\varepsilon}_i \quad (۳۶)$$

$\vec{p}_i$ موقعیت فعلی است، d مقدار جابجایی بلوک سنگ است (رابطه (۳۲))، x مقدار حرکت کارگر است (رابطه (۳۵))، و $\vec{\varepsilon}_i$ یک بردار تصادفی است که توزیع یکنواخت، دارد. گاهی در حین ساخت اهرام، کارگر توانایی خود را از دست می‌داد یا قدرت خود را از دست می‌داد و در نتیجه دیگری جایگزین او می‌شد. این عملیات جایگزینی در الگوریتم با احتمال پنجاه درصد (به طور پیش فرض) انجام می‌شود. بنابراین، پنجاه درصد احتمال دارد که یک کارگر با دیگری در هر تکرار جایگزین شود. استفاده از عملیات جایگزینی بسیار شبیه به اپراتور متقاطع یکنواخت است. فرض می‌شود اگر راه‌حل‌های اولیه مسئله $= (\varphi_1, \varphi_2, , \ldots, \varphi_n)$ باشد و راه‌حل‌های تولید شده با استفاده از رابطه (۳۶)، $\psi = (\psi_1, \psi_2, , \ldots, \psi_n)$ با احتمال پنجاه درصد باشد، برخی از راه حل های اولیه با راه حل‌های تولید شده جایگزین می شوند. بنابراین راه حل‌های جدیدی $Z = (\xi_1, \xi_2, , \ldots, \xi_n)$ خواهیم داشت.

$$\vec{p} = (\vec{p}_i + d)x\vec{\varepsilon}_i \quad (۳۷)$$

۳-۷. الگوریتم بهینه ساز شاهین آتشین

الگوریتم بهینه سازی شاهین آتشین[1] (FHO) بر اساس رفتار جستجوی شاهین‌های قهوه‌ای در شکار پیشنهاد ارائه شده است (عزیزی و همکاران،۱۴۰۰). این پرندگان به عنوان مکانیزمی برای کنترل و گرفتن طعمه خود، چوب‌های سوزان را برمی‌دارند و در مکان‌های نسوخته دیگر می‌اندازند تا آتش‌های کوچکی برپا کنند. این آتش‌های کوچک طعمه‌ها از جمله جوندگان، مارها و سایر حیوانات را می‌ترسانند

[1] Fire Hawk Optimizer (FHO)

و آن‌ها را مجبور می‌کنند با بیشترین عجله فرار کنند که گرفتن آن‌ها را برای شاهین‌ها بسیار آسان‌تر می‌کند.در ابتدا تعدادی از راه حل‌های کاندید X به عنوان بردارهای موقعیت شاهین آتشین و طعمه تعیین می‌شوند.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_1^1\ x_1^2 \dots x_1^j \dots x_1^d \\ x_2^1\ x_2^2 \dots x_2^j \dots x_2^d \\ \vdots\ \vdots\ \ddots\ \vdots \\ x_i^1\ x_i^2 \dots x_i^j \dots x_i^d \\ \vdots\ \vdots\ \ddots\ \vdots \\ x_N^1\ x_N^2 \dots x_N^j \dots x_N^d \end{bmatrix} \quad (۳۸)$$

$$x_i^j(0) = x_{i,min}^{j\,j\left(\begin{smallmatrix} i=1,2,\dots,N. \\ j=1,2,\dots,d. \end{smallmatrix}\right)} \quad (۳۹)$$

که در آن X_i نشان دهنده راه حل i ام در فضای جستجو است؛ d نشان دهنده بعد مسئله در نظر گرفته شده است. N تعداد کل کاندیداهای راه حل در فضای جستجو است. x_i^j، j امین متغیر تصمیم از i امین راه حل کاندید است. $x_i^j(0)$ موقعیت اولیه کاندیداهای راه حل را نشان می‌دهد. $x_{i,min}^j$ و $x_{i,max}^j$ حداقل و حداکثر کران j امین متغیر تصمیم j از i امین راه حل کاندید هستند. و $rand$ یک عدد تصادفی توزیع شده یکنواخت در محدوده [۰،۱] است. به منظور تعیین مکان‌های شاهین‌های آتش در فضای جستجو، ارزیابی تابع هدف برای کاندیداهای راه حل مسئله بهینه سازی انتخاب شده را در نظر می‌گیرد. برخی از کاندیداهای راه حل با مقادیر تابع هدف بهتر به عنوان شاهین‌های آتش انتخاب می شوند، در حالی که بقیه کاندیداهای راه حل طعمه هستند.

$$PR = \begin{bmatrix} PR_1 \\ PR_2 \\ \vdots \\ PR_K \\ \vdots \\ PR_M \end{bmatrix}, K = 1,2,\dots,m \quad (۴۰)$$

$$FH = \begin{bmatrix} FH_1 \\ FH_2 \\ \vdots \\ FH_l \\ \vdots \\ FH_n \end{bmatrix}, l = 1,2,\dots,n, \quad (۴۱)$$

که در آن PR_Kنشان دهنده‌ی k امین طعمه در فضای جستجو با توجه به تعداد کل شکار (m) است. و FH_l نشان دهنده‌ی l امین شاهین آتش با در نظر گرفتن تعداد کل شاهین آتش (n) در فضای جستجو است. در مرحله بعدی الگوریتم، فاصله کل بین شاهین‌های آتش و طعمه محاسبه می‌شود. در نتیجه

نزدیک‌ترین طعمه به هر پرنده مشخص می‌شود تا قلمرو مؤثر این پرندگان مشخص شود. D_k^l با استفاده از رابطه زیر تعیین می‌شود:

$$D_k^l = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad \begin{cases} l = 1,2,\dots,n \\ k = 1,2,\dots,m \end{cases} \qquad (۴۲)$$

که در آن D_k^l فاصله کلی بین شاهین آتشین l ام و k امین طعمه است. m تعداد کل طعمه‌ها در فضای جستجو است. n تعداد کل شاهین های آتش در فضای جستجو است. و (x1, y1) و (x2, y2) مختصات شاهین آتش و طعمه را در فضای جستجو نشان می‌دهند.

پس از انجام روش ذکر شده برای اندازه‌گیری فاصله کل شاهین‌های آتش و طعمه، قلمرو این پرندگان با نزدیک‌ترین طعمه در اطراف آن‌ها مشخص می‌شود. شاهین‌های آتش با مقدار تابع هدف بهتر، نزدیکترین طعمه را در فضای جستجو برای قلمرو خاص خود انتخاب می‌کنند و قوی‌ترین شاهین‌های آتش موفق‌تر از پرندگان ضعیف‌تر شکار کنند. مرحله بعدی الگوریتم، شاهین‌های آتش چوب های سوزان را از آتش اصلی جمع آوری می‌کنند تا در منطقه انتخاب شده آتش به پا کنند. در این مرحله، هر پرنده یک چوب سوزان را برمی دارد و سپس آن را در قلمرو خاص خود می اندازد تا طعمه را مجبور به فرار عجولانه کند. در همین حال، برخی از پرندگان مشتاق استفاده از چوب‌های سوزان قلمروهای دیگر شاهین‌های آتش هستند. بنابراین، این دو رفتار را می توان به عنوان رویه های به روز رسانی موقعیت در حلقه جستجوی اصلی FHO مورد استفاده قرار داد.

$$F\,H_l^{new} = FH_l + (r_1 \times GB - r_2 \times FH_{Near}),\ l = 1,2,\dots,n, \qquad (۴۳)$$

که در آن FH_l^{new} بردار موقعیت جدید l امین شاهین‌های آتش است (FH_l). GB بهترین راه‌حل سراسری در فضای جستجو است که به عنوان آتش اصلی در نظر گرفته می‌شود. FH_{Near}یکی از دیگر شاهین‌های آتش در فضای جستجو است. و r_1 و r_2 اعداد تصادفی به طور یکنواخت در محدوده (۰، ۱) است که برای تعیین حرکات شاهین‌های آتش به سمت آتش اصلی و تعیین مناطق قلمروی سایر شاهین‌های آتش استفاده می‌شوند. در مرحله بعدی الگوریتم، حرکت طعمه در داخل قلمرو هر شاهین آتش یک جنبه کلیدی از رفتار حیوانات برای فرآیند به‌روزرسانی موقعیت در نظر گرفته می‌شود. هنگامی که یک چوب در حال سوختن توسط یک شاهین آتش رها می‌شود، طعمه تصمیم می‌گیرد پنهان شود، فرار کند یا به اشتباه به سمت شاهین آتش می‌دود.

$$PR_q^{new} = PR_q + (r_3 \times FH_l - r_4 \times SP_l) \quad \begin{cases} l = 1,2,\dots,n \\ q = 1,2,\dots,r \end{cases} \qquad (۴۴)$$

که در آن PR_q^{new} بردار موقعیت جدید q امین طعمه است PR_qکه توسط l امین شاهین آتش FH_lاحاطه شده است. GB بهترین راه حل سراسری در فضای جستجو است که به عنوان آتش اصلی در نظر گرفته می‌شود. SP_l مکانی امن در زیر قلمرو l امین شاهین آتش است. و r_3 و r_4 اعداد تصادفی به

طور یکنواخت در محدوده (۰، ۱) برای تعیین حرکات شکار به سمت شاهین آتش و مکان امن توزیع شده‌اند. علاوه بر این، طعمه ممکن است به سمت قلمرو دیگر شاهین‌های آتش حرکت کند در حالی که این احتمال وجود دارد که طعمه‌ها به کمین‌های نزدیک به شاهین‌های آتش نزدیک شوند یا حتی سعی کنند در مکان امن‌تری خارج از قلمرو شاهین‌های آتشی که به دام آن‌ها افتاده‌اند پنهان شوند.

$$PR_q^{new} = PR_q + (r_5 \times FH_{Alter} - r_6 \times SP) \quad \begin{cases} l = 1,2,\dots,n \\ q = 1,2,\dots,r \end{cases} \qquad (۴۵)$$

که در آن PR_q^{new} بردار موقعیت جدید q امین طعمه است PR_qاست که توسط l امین شاهین آتش FH_l احاطه شده است. FH_{Alter} یکی دیگر از شاهین‌های آتش در فضای جستجو است. SP مکانی امن در خارج از قلمرو l امین شاهین آتش است. r_5 و r_6 اعداد تصادفی به طور یکنواخت در محدوده (۰، ۱) برای تعیین حرکات طعمه‌ها به سمت دیگر شاهین‌های آتش و مکان امن خارج از قلمرو آن‌ها، توزیع شده‌اند.

با توجه به اینکه مکان امن در طبیعت مکانی است که اکثر حیوانات برای دور هم جمع شدن در آن جمع می‌شوند تا در هنگام خطر ایمن و سالم بمانند، نمایش ریاضی SP_l و SP به صورت زیر است:

$$SP_l = \frac{\sum_{q=1}^{r} PR_q}{r} \quad \begin{cases} l = 1,2,\dots,n \\ q = 1,2,\dots,r \end{cases} \qquad (۴۶)$$

$$SP = \frac{\sum_{k=1}^{m} PR_k}{m} \quad k = 1,2,\dots,m \qquad (۴۷)$$

که در آن PR_q ، q امین طعمه است که توسط l امین شاهین آتش FH_l احاطه شده است. PR_q ، k امین طعمه در فضای جستجو است.

۳-۸ . الگوریتم بهینه سازی روباه

الگوریتم بهینه‌سازی روباه[1] (FOX) رفتار جستجوی روباه در طبیعت را هنگام شکار طعمه تقلید می‌کند. این الگوریتم مبتنی بر تکنیک‌هایی برای اندازه‌گیری فاصله بین روباه و طعمه‌اش برای اجرای یک پرش کارآمد است (محمد و رشید، ۲۰۲۲)[2] . در ابتدا، FOX جمعیت را مقدار دهی اولیه می‌کند که به آن ماتریس X می گویند. X موقعیت روباه‌های قرمز است. سپس، تناسب هر عامل جستجو با استفاده از توابع معیار استاندارد در هر تکرار محاسبه می‌شود. برای بررسی بهترین تناسب[3] و بهترین موقعیت[4]،

[1] Fox-inspired Optimization Algorithm (FOX)
[2] Mohammed & Rashid, 2022
[3] BestFitness
[4] BestX

ارزش تناسب هر عامل جستجو (هر ردیف در یک ماتریس X) با تناسب عوامل دیگر (ردیف‌های دیگر) مقایسه می‌شود. بهترین تناسب و بهترین موقعیت با استفاده از مقایسه تناسب ردیف جدید $fittness_{i+1}$ و تناسب ردیف قبل $fittness_i$در طول تکرارها بدست می‌آیند. با استفاده از یک متغیر تصادفی به نام r ، احتمال ۵۰٪ را برای انجام اکتشاف یا بهره برداری در FOX اختصاص می‌دهیم. متغیر a برای کاهش عملکرد جستجو با توجه به بهترین موقعیت استفاده می‌شود و پس از هر تکرار، این مقدار کاهش می‌یابد، به این معنی که روباه در هر تکرار طعمه را بهتر دنبال می‌کند. در مرحله بهره برداری، شرایطی داریم که مربوط به شانس کشتن طعمه است که در بخش قبل ذکر شد. مقدار متغیر تصادفی p در محدوده [۰،۱] است. بنابراین، اگر عدد تصادفی p بزرگتر از ۰/۱۸ باشد، باید موقعیت جدید روباه قرمز را پیدا کرد. برای یافتن یک موقعیت جدید، فاصله طی شده توسط صدا $Dist_S_T_{it}$ فاصله روباه قرمز از طعمه $Dist_Fox_\mathrm{Prey}_{it}$ و مقدار پرش $Jump_{it}$باید محاسبه شوند. در نتیجه، یک عدد تصادفی از ۰ تا ۱ برای زمان طی شدن صدا $Time_S_T_{it}$ تولید می‌کنیم. فاصله صوت از روباه قرمز با ضرب سرعت صوت در هوا Sp_S در زمان طی شدن صدا $Time_S_T_{it}$ به دست می‌آید.

$$Dist_S_T_{it} = Sp_S \times Jump \quad (۴۸)$$

که در آن سرعت صوت در محیط Sp_S برابر با ۳۴۳ در هوا و $Time_S_T_{it}$ یک عدد تصادفی در محدوده [۰، ۱] است. it تعداد تکرارها است که از ۱ تا ۵۰۰ متغیر است. با این حال، معادله دیگری برای یافتن Sp_S ساخته شده است، که بر اساس بهترین موقعیت یافت شده تا کنون با تقسیم زمان حرکت صدا بین روباه و طعمه است.

$$Dist_S_T_{it} = Sp_S \times Jump \quad (۴۹)$$

برای یافتن مسافت پیموده شده صوت از رابطه (۴۸)، استفاده می‌شود، در نتیجه فاصله روباه از طعمه $Dist_Fox_\mathrm{Prey}_{it}$ با نصف کردن $Dist_S_T_{it}$ قابل محاسبه است. بنابراین، برای محاسبه فاصله بین حسگر و جسم در فیزیک، فاصله طی شده صوت بر ۲ تقسیم می شود زیرا فاصله سنسور و جسم نصف فاصله‌ای است که توسط موج صوتی پیدا شده است.

$$Dist_Fox_\mathrm{Prey}_{it} = Dist_S_T_{it} \times 0.5 \quad (۵۰)$$

پس از یافتن فاصله بین روباه و طعمه، روباه قرمز باید موقعیت جدیدی پیدا کند تا روباه قرمز برای گرفتن طعمه نیاز به پریدن داشته باشد. بنابراین، روباه باید ارتفاع پرش $Jump_{it}$ را محاسبه کند.

$$Jump_{it} = 0.5 \times 9.81 \times t^2 \quad (۵۱)$$

که در آن ۹/۸۱ شتاب ناشی از گرانش است و t برابر با میانگین زمانی است که صدا طی می‌کند و به دلیل پرش‌های بالا و پایین به صورت مجذور محاسبه می‌شود. رابطه زیر محاسبه موقعیت جدید روباه قرمز را نشان می‌دهد اگر p که عدد تصادفی در محدوده [0,1] است بزرگتر از ۰/۱۸ باشد.

$$X_{(it+1)} = Dist_Fox_\text{Prey}_{it} \times Jump_{it} \times c_1 \quad (۵۲)$$

اما اگر مقدار p کمتر از ۰/۱۸ باشد، موقعیت جدید با رابطه زیر محاسبه می‌شود.

$$X_{(it+1)} = Dist_Fox_\text{Prey}_{it} \times Jump_{it} \times c_2 \quad (۵۳)$$

مقدار c_1 و c_2 به ترتیب ۰/۱۸ و ۰/۸۲ است. این مقادیر بر اساس حرکت پرش یک روباه قرمز است که یا به سمت شمال شرقی یا برعکس می‌پرد. بنابراین، اگر مقدار p بیشتر از ۰/۱۸ باشد، به این معنی است که روباه قرمز به سمت شمال شرقی می‌پرد. در نتیجه، برای یافتن موقعیت جدید، هم $Dist_Fox_\text{Prey}_{it}$ و هم $Jump_{it}$ در c_1 ضرب می‌شوند. بر این اساس، شانس بهره برداری از موقعیت جدید بسیار زیاد است و روباه قرمز به سمت بهینه سراسری می‌رود. با این حال، روباه قرمز در جهت مخالف شمال شرقی می‌پرد، اگر مقدار p کمتر از ۰/۱۸ باشد، این بدان معنی است که شانس کشتن طعمه کم است (۱۸درصد). بنابراین، هر دو $Dist_Fox_\text{Prey}_{it}$ و $Jump_{it}$ در c_2 ضرب می‌شوند. برای کنترل حرکت تصادفی، روباه در این مرحله با توجه به بهترین موقعیت روباه که تاکنون پیدا شده است، به صورت تصادفی جستجو می‌کند. روباه در این مرحله تکنیک پرش ندارد زیرا برای کاوش طعمه در منطقه جستجو، باید به طور تصادفی حرکت کند. برای اطمینان از اینکه روباه به طور تصادفی به سمت بهترین موقعیت حرکت می‌کند، از یک متغیر حداقل زمان $MinT$ و متغیر a برای کنترل جستجو استفاده می‌شود. معادلات (۵۴) و (۵۵) محاسبه $MinT$ و متغیرها را نشان می‌دهد. $MinT$ با یافتن حداقل tt محاسبه می‌شود.

$$tt = \frac{sum\left(Time_{ST_{it}}(i,:)\right)}{\text{dimension}},\ MinT = Min(tt) \quad (۵۴)$$

$sum\left(Time_{ST_{it}}(i,:)\right)$ بر بعد مسئله تقسیم می‌شود تا حداقل میانگین زمانی tt پیدا شود.

$$a = 2 \times \left(it - \left(\frac{1}{Max_{it}}\right)\right) \quad (۵۵)$$

که در آن Max_{it}حداکثر تکرار است. بهترین راه حل $BestX_{it}$ که تا الان پیدا شده تاثیر زیادی در مرحله اکتشاف دارد. رابطه تکنیک اکتشاف روباه را در جستجوی موقعیت جدید در فضای جستجو را نشان می دهد $X_{(it+1)}$.

$$X_{(it+1)} = BestX_{it} \times rand(1,\text{dimension}) \times MinT \times a \quad (۵۶)$$

٤. یافته‌ها

پژوهش ارائه شده از نوع کاربردی بوده و هدف از آن ایجاد چهارچوبی برای پیش‌بینی قیمت رمز ارز بیت کوین[1] در بازه‌های زمانی کوتاه‌مدت (۱۰روزه) و بلندمدت (۳۰ روزه) با استفاده از روش یادگیری ماشین

[1] Bitcoin

به نام شبکه عصبی مصنوعی[1] بهینه سازی شده توسط شش الگوریتم بهینه سازی فراابتکاری به نام‌های الگوریتم PO، HBO، SPO، GPC، FHO و FOX می‌باشد. برای تبیین الگوریتم یادگیری شبکه عصبی، جهت پیشبینی قیمت رمز ارز بیت کوین، مراحل انجام کار به این شرح است:

٤-١. پیش پردازش داده‌ها

مرحله آماده‌سازی داده‌ها قبل از پردازش را، پیش‌پردازش[2] می‌گویند. پیش‌پردازش داده‌ها یک تکنیک یادگیری ماشین شامل تبدیل داده‌های خام به داده‌های قابل فهم و در واقع گام مهمی در فرآیند حل مسائل یادگیری ماشین است. برخی از مهم‌ترین مواردی که طی فرآیند پیش‌پردازش داده‌ها به آنها پرداخته شده است شامل: داده‌های ناموجود[3]، داده‌های پرت[4]، نرمال‌سازی داده‌ها[5] و تقسیم‌بندی داده‌ها[6] می باشد.

٤-١-١. داده‌های ناموجود

در برخی موارد، ممکن است بعضی از ویژگی‌های مربوط به یک یا چند نمونه، فاقد مقادیر معتبر باشند. این امر می‌تواند دلایل مختلفی داشته باشد، از جمله نویزی[7] بودن داده‌های ثبت شده، عدم ثبت و یا نامعتبر بودن مقدار آن. این داده‌ها را داده‌های ناموجود می‌نامند. روش‌های گوناگونی برای مدیریت داده‌های ناموجود وجود دارند که شامل حذف نمونه، پر کردن دستی، استفاده از یک مقدار ثابت، استفاده از مقدار میانگین ویژگی و ... است.

٤-١-٢. داده‌های پرت

در برخی موارد، ممکن است به دلایلی، مقادیری در مجموعه داده ظاهر شوند که تفاوت زیاد و غیر معمولی با سایر مقادیر موجود در مجموعه داشته باشند، این داده‌ها را داده‌های پرت می‌گویند. آنالیز و مدیریت داده‌های پرت یکی از مهمترین مراحل پیش‌پردازش داده‌ها محسوب می‌شود، زیرا این داده‌ها می‌توانند عملکرد الگوریتم‌ها و «دسته‌بندهای[8]» مورد استفاده را دچار مشکل کنند. بنابراین بایستی تا حد ممکن، تاثیر داده‌های پرت را کاهش داد.

[1] Artificial Neural Network (ANN)
[2] Preprocess
[3] Missing Data
[4] Outliers
[5] Normalization
[6] Data Splitting
[7] Noise
[8] Classifier

٤-١-٣. نرمال سازی داده‌ها

برای دستیابی به نتایج دقیق‌تر، لازم است که بازه مربوط به ویژگی‌های مختلف، به نحوی با یکدیگر یکسان و یا نزدیک شوند. برای این منظور از روش‌های نرمال‌سازی استفاده می‌شود. از جمله متداول‌ترین روش‌های نرمال‌سازی می‌توان به روش‌های نرمال‌سازی مین – ماکس[1]، نرمال‌سازی نمره زد[2]، مقیاس گذاری اعشاری[3] و ... اشاره کرد. به منظور کاهش اثرات مقیاس داده‌ها بر روی نتایج مدل، نرمال سازی داده‌ها در بازه ۰ تا ۱ انجام شد. قبل از شروع مدل‌سازی ابتدا بایستی ورودی‌ها و در بعضی از موارد خروجی‌ها را نیز نرمال کرد زیرا وارد کردن داده ها به صورت خام باعث کاهش سرعت و دقت مدل می شود. برای نرمال کردن داده‌ها از رابطه (۵۷) استفاده می‌کنیم، این رابطه داده ها را در بازه a و b نرمال می‌کند. در این رابطه X_{min} ،X_{max} ،XN ، به ترتیب مقدار مینیمم و ماکزیمم داده های ورودی و داده نرمال شده است. همچنین a و b نیز به ترتیب برابر با حد پایین و بالای بازه مورد نظر برای نرمالیزه کردن می‌باشد که در اینجا به ترتیب برابر با ۰ و ۱ می‌باشند.

$$XN = a + \frac{X - X_{min}}{Xmin_{max} \times (b-a)} \quad (۵۷)$$

٤-١-٤. تقسیم‌بندی داده‌ها

در یادگیری ماشین معمولا داده‌ها را به دو قسمت تفکیک می‌کنند. مجموعه‌داده‌های آموزش و آزمایش. در این تحقیق از۸۰ درصد، ابتدای مجموعه داده‌های قیمت بیت کوین به عنوان داده‌های آموزش و ۲۰ درصد باقی‌مانده از انتهای داده‌ها، به عنوان داده‌های آزمایش استفاده شده است.

داده‌های آموزشی[4]: از این بخش از داده‌ها به منظور ایجاد و آموزش مدل‌ها و الگوریتم‌های مختلف یادگیری ماشین و برآورد پارامترهای آن استفاده می‌شود.

داده‌های آزمایشی[5]: این قسمت از داده‌ها برای بررسی کارایی مدل‌ها و الگوریتم‌های مختلف یادگیری ماشین که در مرحله قبل آموزش دیده‌اند، استفاده می‌شود. اهمیت این بخش از داده‌ها در این نکته است که این مشاهدات شامل مقدارهای متغیرهای مستقل (xها) و پاسخی (y) هستند که در آموزش مدل‌های یادگیری ماشین به کار نرفته ولی امکان مقایسه مقدار پیش‌بینی شده توسط مدل‌های یادگیری ماشین را با مقدار واقعی به ما می‌دهند.

[1] Min-Max
[2] Z-Score
[3] Decimal Scaling
[4] Training set
[5] Test set

٤-٢. مدل پیشنهادی پیش بینی قیمت بیت کوین

شبکه عصبی مصنوعی به عنوان یک روش نوین، در مدل سازی و پیش بینی فرآیندها و توابع غیرخطی که برای شناخت و توصیف د قیق آن‌ها راه حل و رابطه صریحی وجود ندارد، عملکرد خوبی از خود نشان داده است. مدل‌های شبکه عصبی مصنوعی با آموزشی که دیده‌اند، می توانند، بدون ایجاد رابطه صریح ریاضی، رفتار سیستم را پیش بینی کنند. از طرفی میزان کارایی شبکه های عصبی مصنوعی به طور شگفت انگیزی به نحوه به روز رسانی وزن‌ها[1] و بایاس‌های[2] آن بستگی دارد. در مقادیر وزن‌های اتصالات نورونی یک شبکه عصبی فرضی سه لایه با ω و مقادیر بایاس‌های آن با β نشان داده شده است.شبکه های عصبی مصنوعی، مدل هایی هستند که توانایی آموزش را با الهام از توانایی یادگیری مغز انسان دارا می باشند. این شبکه‌ها ما را قادر به طراحی سیتم‌های پیچیده و اغلب غیر خطی بین داده‌های ورودی و خروجی کرده و به طور نامحدودی توانایی پذیرش مقدار بسیار بزرگی از ورودی‌ها را دارند. با این وجود این شبکه‌ها دارای محدودیت‌هایی نیز هستند.

استفاده از الگوریتم‌های بهینه سازی فراابتکاری مانند PO، HBO، SPO، GPC، FHO، FOX به علت ماهیت تصادفی بودنشان، می‌تواند تا حد زیادی نقص روش‌های مبتنی بر گرادیان که در مینیمم‌های محلی گیر می‌کنند را برطرف کنند. همچنین این الگوریتم‌ها بر روی انواع شبکه‌های عصبی مصنوعی با هر نوع تابع فعال‌ساز، برای انواع مسائل پیچیده قابل استفاده هستند. به طور کلی آموزش ANN را می‌توان اصلاح مقادیر وزن‌ها و بایاس‌های آن در نظر گرفت که در ابتدا به صورت تصادفی مقداردهی شده‌اند. این وزن‌ها و بایاس‌ها در هر بار تکرار تغییر پیدا می‌کنند و تا زمانی که اختلاف خروجی ANN و خروجی واقعی متناظر با ورودی‌ها، کمتر از یک حد معین شود، اصلاح می‌شوند. پس به روز رسانی وزن‌ها و بایاس‌های ANN یک نوع مسئله بهینه‌سازی است. از طرفی الگوریتم‌های بهینه سازی فراابتکاری مانند PO، HBO، SPO، GPC، FHO، FOX، روش‌های هوشمند جدید و قوی برای یافتن نقاط بهینه در فضای جستجوی مسئله بهینه سازی هستند. بنابراین می‌توان از آن‌ها برای تنظیم وزن‌ها و بایاس‌ها در ANN بهره برد.در نتیجه با توجه به دلایلی که گفته شد برای رفع این نوع از کاستی‌ها و محدودیت‌ها باید به بهینه سازی شبکه‌های عصبی مصنوعی توسط الگوریتم‌های بهینه سازی فراابتکاری پرداخت. بدین منظور در این پژوهش به بهینه سازی شبکه عصبی مصنوعی با استفاده از الگوریتم‌های بهینه سازی فراابتکاری مانند PO، HBO، SPO، GPC، FHO، FOX، به منظور پیش بینی قیمت رمز ارز بیت کوین می‌پردازیم. در این حالت، آموزش شبکه عصبی مصنوعی، روند زمان بر و پرهزینه و اغلب توام با خطای آموزش، با آموزش بدون دخالت خارجی توسط الگوریتم‌های فراابتکاری جایگزین می‌شود.

[1] Weight
[2] Bias

همچنین مراحل تکراری و خسته کننده آزمون و خطا برای رسیدن به خطای مطلوب، حذف می‌شود. این نوع از آموزش با افزایش پیچیدگی شبکه کارایی خود را بهتر نشان می‌دهد.

روند کار بدین صورت است که ابتدا شبکه عصبی مورد نظر را تعریف می‌کنیم. ساختار شبکه بدین صورت است که به تعداد متغیرهای مستقل نورون ورودی داریم. در لایه میانی یا لایه پنهان به تعداد مشخص شده (ثابت) چند نورون تعریف می‌کنیم و در لایه آخر نیز بسته به تعداد خروجی‌ها نورون خروجی داریم. بعد از ایجاد شبکه عصبی مصنوعی، باید وزن‌ها و بایاس‌های آن را به صورت تصادفی مقداردهی کنیم. سپس برای اینکه بتوانیم از الگوریتم‌های فراابتکاری برای آموزش و به روز رسانی وزن‌ها و بایاس‌های شبکه عصبی استفاده کنیم، لازم است که آموزش شبکه عصبی را به عنوان یک تابع (اختلاف میان خروجی‌های واقعی وخروجی‌های شبکه عصبی) در نظر بگیریم و هدفمان بهینه‌کردن (مینیمم‌سازی) آن در یک فضای M بعدی باشد که M برابر با تعداد کل وزن‌ها و بایاس‌های شبکه عصبی ایجاد شده است. حال مسئله بهینه سازی ایجاد شده را می‌توان توسط الگوریتم‌های بهینه‌سازی فراابتکاری حل نمود و با فراخوانی این الگوریتم‌ها وزن‌ها و بایاس‌های شبکه عصبی را بهینه سازی نمود. در هر تکرار الگوریتم، بهترین وزن‌ها و بایاس‌ها انتخاب می‌شوند و در آخر بهترین جواب در بین کل جواب‌ها به عنوان جواب بهینه‌ی الگوریتم برگردانده می‌شود. این مقادیر به عنوان وزن‌ها و بایاس‌های شبکه عصبی در نظر گرفته می‌شوند و وزن‌ها را بروز رسانی می‌کنیم. در انتها شرط پایانی را بررسی کرده و در صورت برقرار بودن الگوریتم خاتمه می‌یابد. شرط پایانی معمولا به دو صورت در نظر گرفته می‌شود: ۱- خطای مینیمم مربعات برای شبکه از یک آستانه کمتر باشد ۲- تعداد تکرارهایی که از قبل مشخص شده باشند. در نهایت با برقرار شدن شرایط پایانی به شبکه عصبی با وزن‌ها و بایاس‌های بهینه دست یافته‌ایم. به منظور آموزش شبکه عصبی مصنوعی با استفاده از الگوریتم‌های بهینه سازی فراابتکاری PO، HBO، SPO، GPC، FHO، FOX، وزن‌ها و بایاس‌های هر نورون مصنوعی در شبکه، در بردارهایی که به صورت تصادفی تولید می‌شوند به گونه ای تعیین می‌شود که شبکه عصبی مصنوعی دارای کمترین خطا بر روی داده‌های آموزشی باشد. طول این بردار برابر است با تعداد کل وزن‌ها و بایاس‌های مورد نیاز برای تولید شبکه عصبی مصنوعی. در واقع به تعداد اعضای جمعیت تولید شده در الگوریتم‌های بهینه سازی فراابتکاری PO، HBO، SPO، GPC، FHO، FOX، بردارهایی تصادفی که هریک شامل مقادیر وزن‌ها و بایاس‌های شبکه عصبی است تولید و سپس توسط تابع هدف RMSE میزان بهینه بودن این شبکه‌های عصبی تولید شده از نظر خطای بین مقادیر واقعی و مقادیر پیش‌بینی شده توسط شبکه عصبی، ارزیابی و در انتهای الگوریتم‌های بهینه‌سازی فراابتکاری PO، HBO، SPO، GPC، FHO، FOX، شبکه عصبی بهینه با کمترین میزان خطا بدست می‌آید.تابع هدف مورد استفاده در شبکه‌های عصبی مصنوعی در رابطه (۵۸) نمایش داده شده است. در این تابع z مقدار واقعی قیمت بیت‌کوین، $\bar{Z}$ مقدار پیش بینی شده قیمت بیت کوین توسط شبکه عصبی و n تعداد کل داده‌ها می‌باشد.

در مساله بهینه سازی وزن‌های شبکه های عصبی مصنوعی، باید ترتیبی از وزن‌ها را بدست آورد که با اعمال آن‌ها به شبکه عصبی مصنوعی، میزان اختلافات میان خروجی واقعی و خروجی شبکه عصبی مصنوعی بر روی داده‌های موجود به کمترین مقدار خود برسد.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{Z}_i)} \quad (۵۸)$$

در این پژوهش سعی شده است،تا به پیش‌بینی قیمت رمز ارز بیت کوین در بازه‌های زمانی کوتاه‌مدت (۱۰روزه) و بلندمدت (۳۰ روزه)، با استفاده از روش شبکه عصبی مصنوعی بهینه سازی شده توسط الگوریتم‌های PO، HBO، SPO، GPC، FHO و FOXپرداخته شود. با توجه به اینکه داده‌های رمز ارز بیت کوین، یک سری زمانی را تشکیل می‌دهند، لذا برای پیش‌بینی میزان قیمت رمز ارز در آینده، متغیرهای ورودی به الگوریتم‌های یادگیری ماشین مانند شبکه‌های عصبی مصنوعی ANN همان مشاهدات گام‌های قبلی از این متغیر است. به عبارت دیگر متغیرهای ورودی به مدل‌های یادگیری ماشین، میزان قیمت رمز ارز در روزهای پیشین یا همان گام‌های پیشین است. به این روند ایجاد تاخیر زمانی بر روی پارامترهای ورودی گفته می‌شود. لذا به منظور پیش‌بینی قیمت رمز ارز، در ابتدا تاخیر زمانی بر روی متغیرهای قیمت رمز ارز بیت کوین در هنگام باز شدن[۱]، قیمت بالایی رمز ارز[۲]، قیمت پایینی رمز ارز[۳]، حجم معاملات[۴] و ارزش بازار[۵] ارز لحاظ می‌شود که به صورت $X - a$ مشخص می‌شود که X پارامتر ورودی و a تعداد روز تاخیر می‌باشد. در این تحقیق ورودی‌های مدل‌های یادگیری ماشین، میزان قیمت رمز ارز بیت کوین در هنگام باز شدن، قیمت بالایی رمز ارز، قیمت پایینی رمز ارز، حجم معاملات و ارزش بازار در ۱، ۲، ۳، ۴ و ۵ روز قبل یعنی $(X-1, X-2, X-3, X-4, X-5)$ است، به عبارتی میزان تاخیرها را ۱، ۲، ۳، ۴ و ۵ روز قبل در نظر گرفته‌ایم و متغیر خروجی میزان متغیر قیمت نهایی رمز ارز[۶] در ۱۰ روز بعد یعنی $X + 10$ است که قصد داریم آن را پیش‌بینی کنیم. به عبارت دیگر میزان متغیر قیمت رمز ارز در ۱۰ روز آینده را تابعی از میزان متغیرهای ورودی در ۱، ۲، ۳، ۴ و ۵ روز قبل در نظر گرفته‌ایم که این عبارت در رابطه زیر نمایش داده شده است. با توجه به اینکه ۵ متغیر ورودی هر کدام ۵ تاخیر بر روی آن‌ها لحاظ شده‌اند، لذا در پژوهش حاضر ۲۵ متغیر ورودی و ۱ متغیر خروجی وجود دارد.

$$X + 10 = f(X-1, X-2, X-3, X-4, X-5) \quad (۵۹)$$

[1] Open
[2] High
[3] Low
[4] Volume
[5] Market Cap
[6] Close

همچنین یک بار نیز متغیر خروجی میزان متغیر قیمت نهایی رمز ارز در ۳۰ روز بعد در نظر گرفته شده است، یعنی $X+30$ که قصد داریم آن را پیش‌بینی کنیم. به عبارت دیگر میزان متغیر قیمت رمز ارز در ۳۰ روز آینده را تابعی از میزان متغیرهای ورودی در ۱، ۲، ۳، ۴ و ۵ روز قبل در نظر گرفته‌ایم که این عبارت در رابطه زیر نمایش داده شده است.

$$X+30=f(X-1,X-2,X-3,X-4,X-5) \quad (۶۰)$$

٤-۳. معیارهای ارزیابی و اعتبارسنجی

در این پژوهش، به منظور ارزیابی و مقایسه کارآیی الگوریتم‌های بهینه سازی، بهینه‌ساز سیاسی PO، بهینه‌ساز مبتنی بر سلسله مراتب رتبه در شرکت HBO، بهینه‌ساز رنگ‌های تصادفی SPO، الگوریتم ساخت اهرام جیزه GPC، بهینه‌ساز شاهین آتشین FHO و الگوریتم بهینه‌سازی فاکس FOX در بهینه سازی شبکه عصبی مصنوعی و پیش‌بینی قیمت رمز ارز، از معیارهای معتبر به شرح موارد زیر استفاده شده است. در این روابط، $x_{imeas}, x_{ipred}, \bar{x}, n$ به ترتیب برابر با تعداد متغیر اندازه‌گیری شده، مقدار میانگین متغیر اندازه‌گیری شده، مقدار متغیر پیش‌بینی شده و مقدار متغیر اندازه‌گیری شده می‌باشد.

٤-۳-۱. ضریب تعیین

ضریب تعیین[1] که با R^2 نمایش داده می‌شود، قدرت توضیح دهندگی مدل را نشان می‌دهد. ضریب تعیین نشان می‌دهد که چند درصد از تغییرات متغیر وابسته توسط متغیرهای مستقل توضیح داده می‌شود. ضریب تبیین بیان کننده بخشی از کل واریانس مقادیر مشاهده‌ای است که توسط مقادیر شبیه‌سازی شده توجیه می شود. ضریب تبیین بین صفر تا یک تغییر می‌کند و مقدار بهینهٔ آن یک است و این حالت مربوط به زمانی است که در آن مقادیر شبیه سازی شده دقیقا مشابه مقادیر مشاهداتی باشد.

$$R^2=1-\frac{\sum_{i=1}^{n}(x_{imeas}-x_{ipred})^2}{\sum_{i=1}^{n}(x_{imeas}-\bar{x})^2} \quad (۶۱)$$

٤-۳-۲. میانگین مربعات خطا

میانگین مربعات خطا[2] (MSE) روشی برای برآورد میزان خطاست که در واقع تفاوت بین مقادیر تخمینی و آنچه تخمین زده شده، است. این شاخص که مقداری همواره نامنفی دارد، هرچقدر مقدار آن به صفر نزدیکتر باشد، نشان دهنده میزان کمتر خطاست. مقدار این شاخص به صورت زیر بیان می شود:

$$MSE=\frac{1}{n}\times\sum_{i=1}^{n}[(x_{imeas}-x_{ipred})^2] \quad (۶۲)$$

[1] Determination Coefficient
[2] Mean Squared Error (MSE)

٤-٣-٣. مجذور میانگین مربعات خطا

مجذور میانگین مربعات خطا[1] (RMSE) در واقع مجذور شاخص میانگین مربعات خطاست. این شاخص به عنوان معیاری از خطای مطلق بین متغیر شبیه سازی و مشاهده‌ای است. مقدار این شاخص آماری بین صفر تا بی نهایت متغیر است. هر چه مقدار این شاخص کمتر باشد شبیه سازی بهتری صورت گرفته است و مقدار بهینهٔ آن صفر است.

$$RMSE = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} [(x_{imeas} - x_{ipred})^2]} \quad (۶۳)$$

٤-٣-٤. میانگین قدر مطلق خطا

میانگین قدر مطلق خطا[2] (MAE) میانگین اختلاف بین مقدار پیش‌بینی شده و مقدار واقعی در تمام موارد آزمایش است. این خطا خطای متوسط پیش بینی است. هر چه مقدار این شاخص کمتر باشد شبیه سازی بهتری صورت گرفته اس-ت و مق-دار بهین-هٔ آن صفر است.

$$MAE = \frac{1}{n} \times \sum_{i=1}^{n} |x_{imeas} - x_{ipred}| \quad (۶۴)$$

٤-٣-٥. مربع خطای نسبی

مربع خطای نسبی[3] (RSE) نشان‌دهنده‌ی عدم تناسب مدل با داده‌های موجود است. به عبارت ساده‌تر، اگر مقدار RSE به صفر نزدیک باشد، مدل شما به خوبی متناسب با داده ها است. در غیر این صورت آنگاه مدل دارای تناسب خوبی با داده‌ها نمی‌باشد.

$$RSE = \frac{\sum_{i=1}^{n} (x_{ipred} - x_{imeas})^2}{\sum_{i=1}^{n} (\bar{x}_{meas} - x_{imeas})^2} \quad (۶۵)$$

٤-٣-٦. واریانس تبیین شده

در آمار، واریانس تبیین شده[4] (EVS)، تناسبی را که یک مدل ریاضی برای تغییرات (پراکندگی) یک مجموعه داده معین انجام می‌دهد، اندازه گیری می کند. بهترین مقدار ممکن برای این معیار ۱ است، مقادیر پایین بدتر هستند.

$$EVS = 1 - \frac{Var\{x_{imeas} - x_{ipred}\}}{Var\{x_{imeas}\}} \quad (۶۶)$$

[1] Roost Mean Squared Error (RMSE)
[2] Mean Absolute Error (MAE)
[3] Relative Squared Error (RSE)
[4] Explained Variance Score

٤-٤. پیش بینی قیمت رمز ارز بیت کوین در بازه زمانی کوتاه مدت ۱۰ روز آینده

پیش‌بینی قیمت رمز ارز توسط شبکه عصبی مصنوعی بهینه سازی شده توسط الگوریتم‌های بهینه‌سازی PO، HBO، SPO، GPC، FHO و FOX در بازه‌ی زمانی کوتاه‌مدت ۱۰روزه، در نرم افزار MATLAB انجام شد که نتایج و معیارهای ارزیابی در دو حالت آموزش و آزمایش به شرح زیر می‌باشند.

جدول (۲): نتایج پیش‌بینی قیمت رمز ارز بیت کوین در بازه زمانی کوتاه مدت ۱۰ روز آینده

پارامتر	مرحله	R^2	RMSE	MSE	MAE	RSE	EVS
ANN-PO	Train	۰/۹۵۶۱	۰/۰۰۰۶۲۱۵	۰/۰۲۴۹۳	۰/۰۰۹۶۵۹۷	۰/۰۴۳۹۶۳	۰/۷۹۰۳۸
	Test	۰/۸۳۰۷۹	۰/۰۱۱۵۲۶	۰/۱۰۷۳۶	۰/۰۹۲۷۶۸	۰/۳۴۴۸۵	۰/۵۸۴۶۸
ANN-HBO	Train	۰/۹۴۲۱	۰/۰۰۰۹۹۹۳۲	۰/۰۳۱۶۱۲	۰/۰۱۷۲۲۲	۰/۰۷۰۶۸۸	۰/۷۴۰۸۵
	Test	۰/۸۰۹۱۳	۰/۰۱۹۶۷۸	۰/۱۴۰۲۸	۰/۱۱۶۲۹	۰/۵۸۸۷۲	۰/۵۳۱۸۲
ANN-SPO	Train	۰/۹۷۳	۰/۰۰۰۳۸۱۷۸	۰/۰۱۹۵۳۹	۰/۰۰۷۴۱۵۹	۰/۰۲۷۰۰۵	۰/۸۳۵۶۷
	Test	۰/۸۶۳۸۳	۰/۰۰۹۵۸۶۴	۰/۰۹۷۹۱	۰/۰۸۲۸۱۸	۰/۲۸۶۸۱	۰/۶۲۹۶۸
ANN-GPC	Train	۰/۹۵۹۷۲	۰/۰۰۰۸۴۲۹۸	۰/۰۲۹۰۳۴	۰/۰۰۹۰۹۸۳	۰/۰۵۹۶۲۹	۰/۷۵۸۰۸
	Test	۰/۸۸۰۹۵	۰/۰۰۹۴۹۱۳	۰/۰۹۷۴۲۳	۰/۰۶۶۳۱۴	۰/۲۸۳۹۶	۰/۵۴۸۴
ANN-FHO	Train	۰/۹۶۲۲	۰/۰۰۰۶۶۰۸۷	۰/۰۲۵۷۰۷	۰/۰۱۰۶۱۹	۰/۰۴۶۷۴۸	۰/۷۸۴۷۳
	Test	۰/۸۴۱۲۵	۰/۰۰۵۷۶۲۶	۰/۰۷۵۹۱۲	۰/۰۵۶۱۰۹	۰/۱۷۲۴۱	۰/۵۸۶۱۹
ANN-FOX	Train	۰/۹۵۲۴۶	۰/۰۰۰۶۷۳۹۹	۰/۰۲۵۹۶۱	۰/۰۱۰۷۰۲	۰/۰۴۷۶۷۶	۰/۷۸۱۷۴
	Test	۰/۷۹۵۹۴	۰/۰۱۱۷۷	۰/۱۰۸۴۹	۰/۰۹۴۴۴۷	۰/۳۵۲۱۴	۰/۵۲۹۹۹

منبع: یافته های تحقیق

٤-٥. پیش بینی قیمت رمز ارز بیت کوین در بازه زمانی بلند مدت ۳۰ روز آینده

پیش‌بینی قیمت توسط الگوریتم‌های بهینه سازی PO، HBO، SPO، GPC، FHO و FOX در بازه‌ی زمانی بلندمدت ۳۰ روزه انجام شد که نتایج و معیارهای ارزیابی در دو حالت آموزش و آزمایش به شرح زیر می‌باشند.

جدول (۳): نتایج پیش‌بینی قیمت رمز ارز بیت کوین در بازه زمانی بلند مدت ۳۰ روز آینده

پارامتر	مرحله	R^2	RMSE	MSE	MAE	RSE	EVS
ANN-PO	Train	۰/۹۱۰۶۶	۰/۰۰۱۳۳۲۸	۰/۰۳۶۵۰۷	۰/۰۱۸۹۲	۰/۰۸۹۸۷۷	۰/۷۰۰۲۴
	Test	۰/۷۰۷۸۷	۰/۰۱۰۷۷۸	۰/۱۰۳۸۲	۰/۰۸۳۱۱	۰/۳۲۶۳۷	۰/۴۵۲۷۹
ANN-HBO	Train	۰/۹۱۵۲۶	۰/۰۰۱۳۵۶۵	۰/۰۳۶۸۳۱	۰/۰۱۴۹۵۲	۰/۰۹۱۴۷۸	۰/۶۹۷۶۶
	Test	۰/۶۶۲۷۸	۰/۰۱۱۳۱۳	۰/۱۰۶۳۶	۰/۰۸۰۶۴۷	۰/۳۴۲۵۶	۰/۴۱۹۲۵

پارامتر	مرحله	R^2	RMSE	MSE	MAE	RSE	EVS
ANN-SPO	Train	۰/۹۴۸۸۹	۰/۰۰۰۷۵۸۴۹	۰/۰۲۷۵۴۱	۰/۰۱۲۹۰۱	۰/۰۵۱۱۵	۰/۷۷۳۸۵
	Test	۰/۶۹۹۱۷	۰/۰۳۱۹۹۳	۰/۱۷۸۸۷	۰/۱۵۲۵	۰/۹۶۸۷۶	۰/۴۱۴۲۵
ANN-GPC	Train	۰/۹۳۱۵۶	۰/۰۰۱۰۳۰۳	۰/۰۳۲۰۹۸	۰/۰۱۴۷۶۵	۰/۰۶۹۴۷۷	۰/۷۳۶۸۴
	Test	۰/۷۲۱۱۴	۰/۰۱۶۰۶۹	۰/۱۲۶۷۶	۰/۱۰۳۳۷	۰/۴۸۶۵۷	۰/۴۵۴۵۵
ANN-FHO	Train	۰/۹۲۹۶	۰/۰۰۱۰۹۲۵	۰/۰۳۳۰۵۳	۰/۰۲۰۳۰۷	۰/۰۷۳۶۷۳	۰/۷۳۳۸۸
	Test	۰/۶۹۸۸۵	۰/۰۱۵۲۶۱	۰/۱۲۳۵۳	۰/۰۹۵۱۹۲	۰/۴۶۲۱	۰/۴۰۵۳۶
ANN-FOX	Train	۰/۹۴۸۶۸	۰/۰۰۰۷۸۲۵	۰/۰۲۷۹۷۳	۰/۰۱۵۳۷۵	۰/۰۵۲۷۶۹	۰/۷۷۲۸۹
	Test	۰/۷۰۷۲۸	۰/۰۲۴۲۴۹	۰/۱۵۵۷۲	۰/۱۲۴۶۴	۰/۷۳۴۲۸	۰/۳۸۱۰۹

منبع: یافته های تحقیق

۵. نتیجه‌گیری و پیشنهادات

پیش‌بینی قیمت بیت کوین، به دلایل متعددی حائز اهمیت است. با پیش‌بینی قیمت، سرمایه‌گذاران می‌توانند تصمیمات بهتری برای مدیریت ریسک‌های مالی خود اتخاذ کنند و از ضررهای احتمالی جلوگیری نمایند. تحلیل دقیق روندهای بازار به سرمایه‌گذاران کمک می‌کند تا استراتژی‌های مناسب برای خرید و فروش را طراحی کنند. شبکه‌های عصبی می‌توانند با پردازش حجم بالای داده‌ها و یادگیری از آن‌ها، پیش‌بینی‌هایی با دقت بالا ارائه دهند که به سرمایه‌گذاران کمک می‌کند تصمیمات بهتری بگیرند. در این مقاله به منظور پیش‌بینی قیمت رمز ارز بیت کوین در بازه‌های زمانی کوتاه‌مدت ۱۰ روزه و بلندمدت ۳۰ روزه، از مدل‌های شبکه عصبی مصنوعی بهینه سازی شده توسط الگوریتم‌های PO، HBO، SPO، GPC، FHO و FOX استفاده شده است. ابتدا داده‌های مورد نیاز برای مدل‌سازی را پیش پردازش کرده و مقادیر خالی و ناموجود آن را حذف می‌کنیم. سپس به منظور کاهش اثرات مقیاس داده‌ها بر روی نتایج مدل به نرمال سازی داده‌ها در بازه ۰ تا ۱ می‌پردازیم زیرا وارد کردن داده‌ها به صورت خام باعث کاهش سرعت و دقت مدل می‌شود. سپس با تقسیم بندی داده‌ها از ۸۰ درصد از ابتدای مجموعه داده‌های قیمت بیت کوین به عنوان داده‌های آموزش و ۲۰ درصد باقی‌مانده از انتهای داده‌ها، به عنوان داده‌های آزمایش اقدام به تولید مجموعه داده‌های آموزش و آزمایش می‌کنیم. سپس در مرحله بعد با استفاده از دیتاست آموزش، به آموزش شبکه عصبی مصنوعی ANN با الگوریتم‌های بهینه‌سازی فراابتکاری PO، HBO، SPO، GPC، FHO و FOX می‌پردازیم. سپس در مرحله بعد با استفاده از دیتاست آزمایش، به آزمایش مدل‌های شبکه عصبی مصنوعی که در مرحله قبل توسط الگوریتم‌های بهینه‌سازی ذکر شده، بهینه‌سازی شده است، می‌پردازیم. در مرحله آخر نیز به مقایسه نتایج و میزان دقت و کارآیی مدل‌های مختلف شبکه عصبی مصنوعی در پیش‌بینی قیمت بیت کوین در حالت آموزش و آزمایش می‌پردازیم. معیارهای ارزیابی و اعتبارسنجی مدل‌ها شامل R^2، RMSE، MSE،

MAE، RSE وEVS می‌باشند. تمام مراحل مدل‌سازی در نرم افزار MATLAB R2023b صورت گرفته است. نتایج این تحقیق نشان می‌دهد . نتایج نشان می‌دهد که در بازه زمانی کوتاه مدت ۱۰ روزه و بلند مدت ۳۰ روزه، عملکرد پیش بینی شبکه عصبی بهینه‌سازی شده توسط الگوریتم GPC به ترتیب با ضریب تعیین ۰/۸۸ و ۰/۷۲ بهتر از سایر الگوریتم‌ها بوده است.

٦. تعارض منافع

هیچگونه تعارض منافع توسط نویسندگان بیان نشده است.

References

- Andersen, T. (2018). Intergenerational conflict and public sector size and structure: A rationale for debt limits? Department of Economics and Business Economics, Aarhus University, PeRCent, CESifo, CEPR and IZA, 1-35. Retrieved from https://doi.org/10.1016/j.ejpoleco.2018.07.005/
- Ashrafi, Y., Salimifar, M., Adeli, M.H. and Tavakolian, H. (2018). Investigating the effects of government expenditures on welfare in Iran: Application of dynamic stochastic general equilibrium models. Quartery Journal of Economic Research and Policies, 26(85), 33-82. (In Persian)
- Barro, R. J. (1989). The Ricardian approach to budget deficits. Journal of Economic Perspectives, 3(2), Retrieved from https://doi.org/10.1257/jep.3.2.37/
- Baum, Ch. and Schaffer, M. (2013). A general approach to testing for autocorrelation. Stata Conference, New Orleans, 1-44.
- Centre for the Study of Living Standards. (2021). The index of economic well-being for New Brunswick, 1981-2019. Retrieved from http:// https://www.csls.ca/reports/csls2021-08.pdf
- Emamimibodi, M., Samati, M. and Sharifi renani, H. (2021). The Effect of Fiscal Policy on Social Welfare due to Government Spending Shocks, Monetary and Productivity. Journal of Economic Research and Policies, 29(97), 199-225. Retrieved from https://doi.org/ 10.52547/qjerp.29.97.199/ (In Persian)
- Greiner, A. (2012). Debt and growth: Is there a non-monotonic relation. Working Papers in Economics and Management, 04-2012, Bielefeld University, 1-7.
- Gulley, O. D. (1994). An empirical test of the effects of government deficits on money demand. Applied Economics, 26(3), 239-247. Retrieved from https://doi.org/10.1080/00036849400000005/
- Hamilton, J. D. (1994). Time series analysis. Princeton University Press, 291-336.
- Horry, H., Jalae, S. A. and Lashkari, M. (2020). Investigation the effect of business cycle on the index of economic well-being in Iran. Iranian Journal of Economic Research, 25(82), 149-172. Retrieved from https://doi.org/10.22054/ijer.2020.11913/ (In Persian)

- Italo, G. and Giuseppe, T. (2022). Monetary and Fiscal Policy in a Non-linear model of Public Debt. Working Papers ECONIS- online Catalogue of ZBW.
- Jang, J.H, Puzzello, D. and Zhang, C. (2023). Inflation, Output and Welfare in the Laboratory. Europen Economic Review, 125,104351. Retrieved from https://doi.org/10.1016/j.euroecorev.2022.104351/
- Li, S. and Lin, S. (2022). Housing Property Tax, Economic Growth and Intergenerational Welfare: The Case of China. International Review of Economics and Finance, 83. Retrieved from https://doi.org/10.1016/j.iref.2022.07.010/
- Liaqat, Z. (2019). Does government debt crowd out capital formation? A dynamic approach using panel VAR. Economics Letters, 178(C), 86-90. Retrieved from https://doi.org/10.1016/j.econlet.2019.03.002/
- Lütkepohl, H. (2005). New introduction to multiple time series analysis. ISBN: 978-3-540-27752-1.
- Mousavinik, S. and Bageripormehr, S. (2019). Constructing Government Timeseries Debts and Estimating of the Optimal Ratio of Government Debt to Gross Domestic Product and Financial Space in Iran Economy. The Economic Research, 19(1), 29-52. (In Persian)
- Mowlaei, M., and Golkhandan, A. (2013). Long-Run and Short-Run effects of budget deficit on economic growth in Iran. Quarterly Journal of Economic Strategy, 2(5), 95-115. (In Persian)
- Nademi, Y. and Moftakhari, A. (2017). The optimum government activities and social welfare in Iran during the period of 1975-2012. Social Welfare Quarterly, 17(67), 9-35. (In Persian)
- Ndoricimpa, A. (2017). Analysis of asymmetries in the tax-spending Nexus in Burundi. Journal of Economics and Political Economy, 4(1), 53-70. Retrieved from https://doi.org/10.1453/jepe.v4i1.1160/
- Osberg, L. and Sharpe, A. (2001). The index of economic well-being: An overview. Revised version of a paper presented at the National Conference on Sustainable Development Indicators organized by the National Round Table Environment and the Economy, Westim Hotel, Ottawa, Ontario, 3-10. Retrieved from https://doi.org/10.2753/0577-5132530402/
- Panizza, U. and Presbitero, A. F. (2013). Public debt and economic growth in advanced economies: A survey. Swiss Journal of Economics and Statistics, 175-204. (149), 175-204. Retrieved from https://doi.org/10.1007/BF03399388/
- Rubaie, Q. L. A. and Ahmed, A. S. (2021). Measuring and analyzing the repercussions of public debt in financing the general budget deficit for the Iraqi Economy after 2003 using the (Eviews) Program. Nahrain University, Iraq, Materials today: Proceedings, 80, Part 3, 3144-3154. Retrieved from https://doi.org/10.1016/j.matpr.2021.07.180/
- Sabermahani, M., Zeinaladeh, R., Jalaee Sfandabadi, S. and Zayanderoody, M. (2023). Investigate the Shocks of Real Sector of the Economy on the Welfare

Index in Iranian Economy. Social Eelfare Quarterly, 22(87), 105-148. Retrieved from https://doi.org/10.32598/refahj.22.87.4008.1/ (In Persian)

- Srebrny, J. P. (2021). Not all government budget deficits are created equal: Evidence from advanced economies' sovereign bond markets. Journal of International Money and Finance, 118, 102460. Retrieved from https://doi.org/10.1016/j.jimonfin.2021.102460/
- Westphal, C. Ch. and Ždarek, V. (2017). Fiscal reaction function and fiscal fatigue: evidence for the Euro Area. European Central bank, Working Paper Series, No, 2036, 1-34.
- Zaroki, S., Nasrnejad Nesheli, S. and Gorgani Firoozjah, N. (2023). Exploring Role of Inflation Across Different Commodity Groups on Economic Welfare in Iran. Journal of Economic Research, 23(88), 158-202. Retrieved from https://doi.org/ 10.22054/joer.2024.75021.1148/ (In Persian)