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, monetary economics, faculty of economic, Science and Research Branch, Tehran, Iran

2. Assistant professor, faculty of economics, Science and Research Branch, Tehran, Iran

3. Assistant professor, faculty of economics, Science and Research Branch, Tehran, Iran

4. Assistant professor, faculty of economics, Science and Research Branch, Tehran, Iran

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.

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

* Email: k.peykarjou@srbiau.ac.ir

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

آیدین ابوطالبی[1]، کامبیز پیکارجو[2*]، ابراهیم رضائی[3]، رحیم خانی‌زاد[4]

1. دانشجوی دکتری، گروه اقتصاد، گرایش اقتصاد پولی، دانشکده اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران
2. استادیار، گروه اقتصاد، دانشکده اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران
3. دانشیار، گروه اقتصاد، دانشکده اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران
4. استادیار، گروه اقتصاد، دانشکده اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران

چکیده

رمز ارزها، ارزهای دیجیتالی مبتنی بر فناوری بلاک‌چین 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. بیت‌کوین، 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 کند. برای موفقیت در بازارهای مالی، تخصص و آگاهی ضروری است؛ لذا سرمایه‌گذاران همواره به دنبال روش‌هایی برای پیش‌بینی دقیق‌تر قیمت رمزارزها، به‌ویژه بیت کوین، هستند. با پیشرفت کامپیوترها و افزایش قدرت پردازشی 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 بیت کوین در بازه‌های زمانی کوتاه‌مدت (10روزه) و بلندمدت (30 روزه) با استفاده از شبکه عصبی مصنوعی بهینه سازی شده توسط شش الگوریتم بهینه‌سازی فراابتکاری به نام‌های الگوریتم بهینه ساز سیاسی (PO)، الگوریتم سلسله مراتب رتبه در شرکت (HBO)، الگوریتم بهینه ساز رنگ‌های تصادفی (SPO)، الگوریتم ساخت اهرام جیزه (GPC)، الگوریتم بهینه ساز شاهین آتشین (FHO) و الگوریتم بهینه ساز روباه (FOX) پرداخته شده و نتایج این شش حالت با یکدیگر مقایسه شده است. نتایج نشان می‌دهد که در بازه زمانی کوتاه مدت 10 روزه و بلند مدت 30 روزه، عملکرد پیش بینی شبکه عصبی بهینه‌سازی شده توسط الگوریتم GPC به ترتیب با ضریب تعیین 0/88 و 0/72 بهتر از سایر الگوریتم‌ها بوده است.

کلیدواژه‌ها: پیش‌بینی قیمت رمز ارز، شبکه عصبی مصنوعی، بیت کوین، الگوریتم‌های بهینه‌سازی فراابتکاری

* نویسنده مسئول: Email: k.peykarjou@srbiau.ac.ir

1 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 (Choithani et al., 2024). شبکه‌های عصبی مصنوعی می‌توانند خود را با تغییرات سریع بازار تطبیق دهند و الگوهای جدید را شناسایی کنند (Tayib & Abdulazeez, 2024). استفاده از هوش مصنوعی می‌تولند 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 (Chhajer, Shah, & Kshirsagar, 2022). با توجه به موارد ذکر شده در این پژوهش به پیش‌بینی قیمت رمز ارز بیت کوین در بازه‌های زمانی کوتاه‌مدت 10روزه و بلندمدت 30 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.

2 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 (1402)، در مقاله خود، مدل هیبریدی CEEMD-DL(LSTM) با ترکیب تکنیک‌های یادگیری عمیق و روش تجزیه مد تجربی یکپارچه کامل (CEEMD) 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 2013/01/01 تا 2022/05/28 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 بیت‌کوین شده است. بنابراین، پیشنهاد می‌شود که این مدل برای پیش‌بینی در این زمینه مورد استفاده قرار گیرد (سکینه صیادی, علی اسماعیل, & محمدرضا, 1402).

دووتا و همکاران (2020)، 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 (Dutta, Kumar, Basu, & management, 2020).

سوی و همکاران (2024) 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.

آییلدیز و همکاران (2024) 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.

3 روش تحقیق

در این بخش ابتدا به معرفی مجموعه داده‌های مورد اســتفاده در این پژوهش پرداخته می‌شود، سپس روش یادگیری ماشین شبکه عصبی مصنوعی و شش الگوریتم بهینه‌سازی فراابتکاری به منظور بهینه سازی شبکه‌های عصبی مصنوعی مورد استفاده در این مطالعه شرح داده می‌شود.

1-3 مجموعه داده‌ی مورد استفاده و آماده سازی داده‌ها

مجموعه داده‌های قیمت بیت کوین مورد اســتفاده در این تحقیق مربوط به یک دوره‌ی 13 ساله از 13 جولای 2010 تا 6 دسامبر 2023 جمع آوری و مورد استفاده قرار گرفته است. این مجموعه داده شامل 4895 نمونه و 6 متغیر می‌باشند. متغیرهای قیمت رمز ارز بیت کوین در هنگام باز شــدن[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 1: مقادیر آمار توصیفی ویژگی‌های مجموعه داده قیمت رمز ارز بیت کوین

variables	Open	High	Low	Volume	Market Cap	Close
count	4895	4895	4895	4895	4895	4895
mean	9928.42	10163.18	9676.34	11354598485.61	185343878625.27	9936.44
std	14860.99	15219.79	14463.15	17618890680.47	282008743694.57	14865.68
min	0.06	0.06	0.05	0.00	190260.00	0.06
25%	220.01	226.26	214.80	150985.50	3050629883.50	219.98
50%	1172.71	1191.10	1156.44	315108000.00	19101427518.00	1175.95
75%	11596.48	11898.65	11300.30	19473847922.50	212085926780.50	11600.79
max	67549.74	68789.63	66382.06	350967941479.00	1274831490851.00	67566.83

2-3 شبکه عصبی مصنوعی

شبکه‌های عصبی مصــنوعی 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 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 (Smith, 1993). در حالت کلی در شبکه‌های عصبی، سه لایه‌ی نرونی وجود دارد (Hornik,

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

Stinchcombe, & White, 1989). 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 (Hornik et al., 1989).

در یک شبکه عصبی مصنوعی، نورون mام اطلاعات ورودی خود را از طریق گره های 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 . مقدار 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 (محمدباقر, 1381).

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

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 (محمدباقر, 1381).

$$O_m = f(\sum_{i=1}^{n} I_i W_{i,m} + b_m) \qquad (2)$$

هدف از به کار بردن این تابع، محدود کردن خروجی در یک باند مشخص است. به عبارت دیگر با وجود این تابع، خروجی یک نورون در مقابل ورودی‌ها ی بسیار کوچک یا بسیار بزرگ از حدود معینی تجاوز نمی‌کند (محمدباقر, 1381).

3-3 الگوریتم بهینه‌ساز سیاسی

الگوریتم بهینه ساز سیاسی[1] (PO) در سال 2020 توسط عسکری و همکاران ابداع شد (Askari, Younas, & Saeed, 2020). 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 (Askari, Younas, et al., 2020). فاز تشکیل حزب و تخصیص حوزه انتخابیه فقط یک بار اجرا می‌شود و بقیه چهار فاز در یک حلقه اجرا می‌شوند.

[1] Political Optimizer (PO)

جمعیت P به n حزب سیاسی تقسیم می‌شود (رابطه (3)). هر حزب P_i شامل n نامزد/عضو است (رابطه (4)). هر عضو j از P_i^j به عنوان یک راه حل بالقوه در نظر گرفته می‌شود که یک بردار d-بعدی است (رابطه (5)). 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.

$$P = \{P_1, P_2, P_3, ..., P_n\} \quad (3)$$

$$P_i = \{P_i^1, P_i^2, P_i^3, ..., P_i^n\} \quad (4)$$

$$P_i^j = [P_{i,1}^j, P_{i,2}^j, P_{i,3}^j, ..., P_{i,d}^j]^T \quad (5)$$

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 (6)) و عضو j ام هر حزب از حوزه انتخابیه jام Cj رقابت می‌کند (رابطه (7)).

$$C = \{C_1, C_2, C_3, ..., C_n\} \quad (6)$$

$$C_i = \{P_1^j, P_2^j, P_3^j, ..., P_n^j\} \quad (7)$$

متناسب‌ترین عضو یک حزب به عنوان رهبر حزب اعلام می‌شود که بلافاصله پس از انتخابات عمومی (انتخابات بین حزبی) تصمیم گرفته می‌شود (رابطه (8)) که در آن P_i^* رهبر حزب i را نشان می‌دهد و $f(P_i^j)$ برازندگی P_i^j را محاسبه می‌کند. مجموعه تمام رهبران حزب با P^* نمایش داده می‌شود (رابطه (9)). 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 (10)).

$$q = \arg\min_{1 \le j \le n} f(P_i^j), \forall i \in \{1, ..., n\} \; P_i^* = P_i^q \quad (8)$$

$$P^* = \{P_1^*, P_2^*, P_3^*, ..., P_n^*\} \quad (9)$$

$$C^* = \{C_1^*, C_2^*, C_3^*, ..., C_n^*\} \quad (10)$$

1-3-3 کمپین انتخاباتی (کاوش و استخراج)

این فاز به نامزدها کمک می‌کند تا عملکرد خود در انتخابات را بهبود بخشند که از روابط (11) و (12) برای به‌روزرسانی موقعیت یک نامزد استفاده می‌کند. بسته به رابطه برازندگی فعلی $f(P_i^j(t))$ یک نامزد با برازندگی قبلی خود $f(P_i^j(t-1))$ ، تصمیم گرفته می‌شود که

آیا موقعیت را با استفاده از رابطه (11) یا رابطه (12) به‌روز کنیم. اگر برازندگی بهبود یابد، سپس از رابطه (11) برای به‌روزرسانی موقعیت استفاده می‌شود، با این حال، اگر برازندگی بدتر شود، سپس از رابطه (12) استفاده می‌شود.

$$p_{i,k}^{j}(t+1)=\begin{cases} m^{*}+r(m^{*}-p_{i,k}^{j}(t)), & \text{if } p_{i,k}^{j}(t-1)\le p_{i,k}^{j}(t)\le m^{*} \text{ or } p_{i,k}^{j}(t-1)\ge p_{i,k}^{j}(t)\ge m^{*} \\ m^{*}+(2r-1)|m^{*}-p_{i,k}^{j}(t)|, & \text{if } p_{i,k}^{j}(t-1)\le m^{*}\le p_{i,k}^{j}(t) \text{ or } p_{i,k}^{j}(t-1)\ge m^{*}\ge p_{i,k}^{j}(t) \\ m^{*}+(2r-1)|m^{*}-p_{i,k}^{j}(t-1)|, & \text{if } m^{*}\le p_{i,k}^{j}(t-1)\le p_{i,k}^{j}(t) \text{ or } m^{*}\ge p_{i,k}^{j}(t-1)\ge p_{i,k}^{j}(t) \end{cases} \quad (11)$$

$$p_{i,k}^{j}(t+1)=\begin{cases} m^{*}+(2r-1)|m^{*}-p_{i,k}^{j}(t)|, & \text{if } p_{i,k}^{j}(t-1)\le p_{i,k}^{j}(t)\le m^{*} \text{ or } p_{i,k}^{j}(t-1)\ge p_{i,k}^{j}(t)\ge m^{*} \\ p_{i,k}^{j}(t-1)+r(p_{i,k}^{j}(t)-p_{i,k}^{j}(t-1)), & \text{if } p_{i,k}^{j}(t-1)\le m^{*}\le p_{i,k}^{j}(t) \text{ or } p_{i,k}^{j}(t-1)\ge m^{*}\ge p_{i,k}^{j}(t) \\ m^{*}+(2r-1)|m^{*}-p_{i,k}^{j}(t-1)|, & \text{if } m^{*}\le p_{i,k}^{j}(t-1)\le p_{i,k}^{j}(t) \text{ or } m^{*}\ge p_{i,k}^{j}(t-1)\ge p_{i,k}^{j}(t) \end{cases} \quad (12)$$

2-3-3 تغییر حزب (تعادل بین کاوش و استخراج)

یک پارامتر تطبیقی λ به نام نرخ تغییر حزب تعریف شده است که از $\lambda_{\max}$ شروع می‌شود و به طور خطی به 0 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 (13) بیان شده است.

$$q=\arg\min_{1\le j\le n} f\left(P_i^j\right) \quad (13)$$

3-3-3 انتخابات (ارزیابی برازندگی)

انتخابات با ارزیابی برازندگی تمام نامزدهای رقابت کننده در یک حوزه انتخابیه و اعلام برنده انجام می‌شود (رابطه (14)). رابطه (14) برنده حوزه انتخابیه j ام (Cj) را نشان می‌دهد. همانطور که قبلاً اشاره شد، رهبران حزب نیز پس از انتخابات با استفاده از رابطه (8) به‌روز می‌شوند.

$$\begin{aligned} q&=\arg\min_{1\le i\le n} f\left(P_i^j\right) \\ \mathrm{c}_j^{*}&=P_q^j \end{aligned} \quad (14)$$

4-3-3 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 (8) و (14) 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.

4-3 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 (Askari, Saeed, & Younas, 2020). مدل ریاضی الگوریتم HBO بر سه اصل استوار است: تعامل بین زیردستان و رئیس فوری آن‌ها، تعامل بین همکاران، و خودکمکی کارکنان (Askari, Saeed, et al., 2020).

1-4-3 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:

$$X_i^k(t+1) = B^k + \gamma\lambda^k \left| B^k - X_i^k(t) \right| \tag{15}$$

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:

$$\lambda^k = 2r - 1 \tag{16}$$

که در آن r یک عدد تصادفی است که بر اساس توزیع یکنواخت از محدوده $[0,1]$ تولید می‌شود. در رابطه (15)، γ یک پارامتر با دقت طراحی شده است که به شرح زیر محاسبه

$$\gamma = \left| 2 - \frac{\left(t - \mathrm{mod}\frac{T}{C} \right)}{\frac{T}{4C}} \right| \tag{17}$$

می‌شود:

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 2 به 0 کاهش می‌یابد و پس از رسیدن به 0، شروع به افزایش دوباره به 2 با

[1] Corporate Rank Hierarchy
[2] Heap-based optimizer (HBO)

تکرارها می‌کند. با این حال، پارامتر C است که تعیین می‌کند چند دور γ در T تکرار کامل خواهد شد.

2-4-3 مدل‌سازی ریاضی تعامل بین همکاران

مقامات با یک رتبه به عنوان همکاران در نظر گرفته شده‌لند. 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.

$$X_i^k(t+1)=\begin{cases} S_r^k+\gamma\lambda^k\left|S_r^k-X_i^k(t)\right|, & f\left(\vec{S}_r\right)<f\left(\vec{X}_i(t)\right) \\ X_i^k+\gamma\lambda^k\left|S_r^k-X_i^k(t)\right|, & f\left(\vec{S}_r\right)\geq f\left(\vec{X}_i(t)\right) \end{cases} \quad (18)$$

3-4-3 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:

$$X_i^k(t+1)=X_i^k(t) \quad (19)$$

4-4-3 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 p1، p2، و p3 تقسیم شده است. انتخاب نسبت p1 به یک عامل جستجو اجازه می‌دهد تا موقعیت خود را با استفاده از رابطه (19) به‌روز کند. حد p1 به شرح زیر محاسبه می‌شود:

$$P_1=1-\frac{t}{T} \quad (20)$$

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 (15) به‌روز کند. حد p2 به شرح زیر محاسبه می‌شود:

$$P_2 = P_1 + \frac{1-P_1}{2} \quad (21)$$

در نهایت، انتخاب p3 نشان‌دهنده به‌روزرسانی موقعیت با استفاده از رابطه (18) است و حد p3 به شرح زیر محاسبه می‌شود:

$$P_3 = P_2 + \frac{1-P_1}{2} = 1 \quad (22)$$

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

$$X_i^k(t+1) = \begin{cases} X_i^k(t) & p \le p_1 \\ B^k + \gamma\lambda^k \left|B^k - X_i^k(t)\right|, & p > p_1 \text{ and } p \le p_2 \\ S_r^k + \gamma\lambda^k \left|S_r^k - X_i^k(t)\right|, & p > p_2 \text{ and } p \le p_3 \text{ and } f\left(\vec{S}_r\right) < f\left(\vec{X}_i(t)\right) \\ X_i^k + \gamma\lambda^k \left|S_r^k - X_i^k(t)\right|, & p > p_2 \text{ and } p \le p_3 \text{ and } f\left(\vec{S}_r\right) \ge f\left(\vec{X}_i(t)\right) \end{cases} \quad (23)$$

5-3 الگوریتم بهینه‌ساز رنگ‌های تصادفی

الگوریتم بهینه ساز رنگ‌های تصادفی (SPO) الهام‌گرفته از هنر است (Kaveh, Talatahari, & Khodadadi, 2022). 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 (Kaveh et al., 2022).

1-5-3 فاز 1: مقداردهی اولیه

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

$$C_{i,0} = C_{\min} + rand.\left(C_{\max} - C_{\min}\right), \quad i = 1,2,3,...,nc \quad (24)$$

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 [0, 1] 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.

2-5-3 فاز 2: ارزیابی، مرتب‌سازی و خوشه‌بندی

نقاشی‌ها به عنوان نتیجه مسئله بر اساس تابع هدف مربوطه به ترتیب افزایشی مرتب

می‌شـوند. در نهایت، آن‌ها به سـه گروه برابر یعنی اصـلی (بهترین)، فرعی (خوب) و سـومی (بدترین) خوشـه‌بندی می‌شـوند، به این ترتیب، نیازی به افزودن پارامترها در الگوریتم برای خوشه‌بندی نیست.

3-5-3 فاز 3: 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 (25)).

$$C_{new,1} = C_i + rand.\left(C_{i+1} - C_{i-1}\right) \quad (25)$$

تکنیک ترکیب مکمل رنگ‌هایی که در مقابل یکدیگر در چرخه قرار دارند به عنوان رنگ‌های مکمل شناخته می‌شوند را ترکیب می‌کند (رابطه (26)).

$$C_{new,2} = C_i + rand.\left(C_{Pi} - C_{Ti}\right) \quad (26)$$

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 (27)).

$$C_{new,3} = C_i + rand.\left(\frac{C_{Pi} + C_{Si} + C_{Ti}}{3}\right) \quad (27)$$

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 (28)).

$$C_{new,4} = C_i + \left(rand_1.C_{Pi} + rand_2.C_{Ti} + rand_3.C_{Si} + rand_4.C_{rand}\right)/4 \quad (28)$$

4-5-3 فاز 4: ارزیابی و به‌روزرسانی

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

5-5-3 فاز 5: بررسی شرایط پایان الگوریتم

پس از مجموعه‌ای از تکرارها، چرخه بهینه‌سازی خاتمه می‌یابد. اگر معیار برآورده نشود،

یک فرآیند جدید برای فاز 2 برنامه‌ریزی می‌شود؛ در غیر این صورت، فرآیند متوقف می‌شود و بهترین راه‌حل گزارش می‌شود

6-3 الگوریتم ساخت اهرام جیزه

الگوریتم ساخت اهرام جیزه[1] (GPC) با الهام از گذشته باستان دارای ویژگی‌های یک الگوریتم فراابتکاری خوب برای مقابله با بسیاری از مسائل است.الگوریتم پیشنهادی با حرکات کارگران و هل دادن بلوک‌های سنگی روی سطح شیب دار کنترل می‌شود (Harifi, Mohammadzadeh, Khalilian, & Ebrahimnejad, 2021). 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 15 درجه است و می‌تواند متغیر باشد. (باستان شناسان بر این باورند که زاویه بین 8 تا 12 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.

$$f_k = \mu_k mg\cos\theta \tag{29}$$

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:

$$-mg\sin\theta - f_k = ma \tag{30}$$

که a شتاب است. با قرار دادن رابطه (29) در رابطه (30)، شتاب بلوک سنگ به سمت بالا در سطح شیب‌دار به دست می‌آید.

$$a = -g\left(\sin\theta + \mu_k\cos\theta\right) \tag{31}$$

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

$$d = \frac{v_0^2}{2g\left(\sin\theta + \mu_k\cos\theta\right)} \tag{32}$$

که d مقدار جابجایی است. همانطور که قبلا ذکر شد g گرانش زمین است که مقدار آن 9.8 است. θ زاویه ای است که رمپ با افق ایجاد می کند. v_0 سرعت اولیه بلوک سنگ است

[1] Giza Pyramids Construction (GPC)

و در الگوریتم با یک عدد تصادفی توزیع شده یکنواخت در هر تکرار تعیین می شود.

$$v_0 = rand(0,1) \tag{33}$$

در واقع، $rand(0,1)$ یک عدد تصادفی بین 0 و 1 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.

$$\mu_k = rand\left[\mu_{k_\min}, \mu_{k_\max}\right] \tag{34}$$

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 (32) 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.

$$x = \frac{v_0^2}{2g\sin\theta} \tag{35}$$

پس از محاسبه تغییرات جابجایی بلوک سنگ و حرکت کارگر از طریق روابط (32) و (35)، از حاصل این دو معادله می توان موقعیت جدیدی به دست آورد. این موقعیت جدید یک راه حل جدید است.

$$\vec{p} = \left(\vec{p}_i + d\right) x\vec{\varepsilon}_i \tag{36}$$

$\vec{p}_i$ موقعیت فعلی است، d مقدار جابجایی بلوک سنگ است (رابطه (32))، x مقدار حرکت کارگر است (رابطه (35))، و $\vec{\varepsilon}_i$ یک بردار تصادفی است که توزیع یکنواخت، عادی یا Lévy 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 $\phi = \left(\varphi_1, \varphi_2, \varphi_3, ..., \varphi_n\right)$ باشد و راه‌حل‌های تولید شده با استفاده از رابطه (36)، $\psi = \left(\psi_1, \psi_2, \psi_3, ..., \psi_n\right)$ با احتمال پنجاه درصد باشد، برخی از راه حل های اولیه با راه

حل های تولید شده جایگزین می شوند. بنابراین راه حل‌های جدیدی $Z=(\xi_1,\xi_2,\xi_3,...,\xi_n)$ خواهیم داشت.

$$\xi_k=\begin{cases}\psi_k, & if\ rand[0,1]\le 0.5\\ \varphi_k, & otherwise\end{cases} \quad (37)$$

7-3 الگوریتم بهینه ساز شاهین آتشین

الگوریتم بهینه سازی شاهین آتشین[1] (FHO) بر اساس رفتار جستجوی شاهین‌های قهوه‌ای در شکار پیشنهاد ارائه شده است (Azizi, Talatahari, & Gandomi, 2023). 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 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.

$$X=\begin{bmatrix}X_1\\X_2\\\vdots\\X_i\\\vdots\\X_N\end{bmatrix}=\begin{bmatrix}x_1^1\ x_1^2\ \dots\ x_1^j\dots\ x_1^d\\x_2^1\ x_2^2\ \dots\ x_2^j\dots\ x_2^d\\\vdots\\x_i^1\ x_i^2\ \dots\ x_i^j\dots\ x_i^d\\\vdots\\x_N^1\ x_N^2\ \dots\ x_N^j\dots\ x_N^d\end{bmatrix} \quad (38)$$

$$x_i^j(0)=x_{i,\min}^j+rand.\left(x_{i,\max}^j-x_{i,\min}^j\right),\begin{cases}i=1,2,...,N.\\j=1,2,...,d.\end{cases} \quad (39)$$

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 [0،1] 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

[1] Fire Hawk Optimizer (FHO)

کاندیداهای راه حل طعمه هستند.

(41) (40)

$$PR = \begin{bmatrix} PR_1 \\ PR_2 \\ \vdots \\ PR_K \\ \vdots \\ PR_M \end{bmatrix},\ K = 1,2,...,m, \qquad FH = \begin{bmatrix} FH_1 \\ FH_2 \\ \vdots \\ FH_l \\ \vdots \\ FH_n \end{bmatrix},\ l = 1,2,...,n,$$

که در آن PR_K نشـان دهنده‌ی k امین طعمه در فضـای جسـتجو با توجه به تعداد کل شـکار (m) است. و FH_l نشان دهنده‌ی l امین شاهین آتش با در نظر گرفتن تعداد کل شاهین آتش (n) در فضـای جسـتجو اسـت. در مرحله بعدی الگوریتم، فاصـله کل بین شـاهین‌های 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 می شود:

$$D_k^l = \sqrt{\left(x_2 - x_1\right)^2 + \left(y_2 - y_1\right)^2} \quad \begin{cases} l = 1,2,...,n \\ k = 1,2,...,m \end{cases} \qquad (40)$$

که در آن D_k^l فاصـله کلی بین شـاهین آتشـین l ام و k امین طعمه اسـت. m تعداد کل طعمه‌ها در فضـای جسـتجو اسـت. n تعداد کل شـاهین های آتش در فضـای جسـتجو اسـت. و (x1, y1) و (x2, y2) مختصات شاهین آتش و طعمه را در فضای جستجو نشان می‌دهند.

پس از انجام روش ذکر شـده برای اندازه‌گیری فاصـله کل شـاهین‌های آتش و طعمه، قلمرو این پرندگان با نزدیک‌ترین طعمه در اطراف آن‌ها مشـخص می‌شـود. شـاهین‌های آتش با مقدار تابع هدف بهتر، نزدیکترین طعمه را در فضای جستجو برای قلمرو خاص خود انتخاب می‌کنند و قوی‌ترین شـاهین‌های آتش موفق‌تر از پرندگان ضـعیف‌تر شـکار کنند. مرحله بعدی الگوریتم، شـاهین‌های آتش چوب های سـوزان را از آتش اصـلی 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 لندازد 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 رسـانی موقعیت در حلقه جسـتجوی اصـلی FHO مورد استفاده قرار داد.

$$FH_l^{new} = FH_l + \left(r_1 \times GB - r_2 \times FH_{Near}\right),\ l = 1,2,...,n, \qquad (41)$$

که در آن FH_l^{new} بردار موقعیت جدید l امین شـاهین‌های آتش اسـت (FH_l). GB

بهترین راه حل سراسری در فضای جستجو است که به عنوان آتش اصلی در نظر گرفته می شود. FH_{Near} یکی از دیگر شاهین‌های آتش در فضای جستجو است. و r_1 و r_2 اعداد تصادفی به طور یکنواخت در محدوده (0، 1) 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.

$$PR_q^{new} = PR_q + \left(r_3 \times FH_l - r_4 \times SP_l\right) \quad \begin{cases} l = 1,2,...,n \\ q = 1,2,...,r \end{cases} \tag{42}$$

که در آن PR_q^{new} بردار موقعیت جدید q امین طعمه است (PR_q) که توسط l امین شاهین آتش (FH_l) احاطه شده است. GB بهترین راه حل سراسری در فضای جستجو است که به عنوان آتش اصلی در نظر گرفته می شود. 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 (0، 1) 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.

$$PR_q^{new} = PR_q + \left(r_5 \times FH_{Alter} - r_6 \times SP\right) \quad \begin{cases} l = 1,2,...,n \\ q = 1,2,...,r \end{cases} \tag{43}$$

که در آن PR_q^{new} بردار موقعیت جدید q امین طعمه است (PR_q) است 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 (0، 1) برای تعیین حرکات طعمه ها 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 حیولنات 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:

$$SP_l = \frac{\sum_{q=1}^{r} PR_q}{r} \quad \begin{cases} l = 1,2,...,n \\ q = 1,2,...,r \end{cases} \tag{44}$$

$$SP = \frac{\sum_{k=1}^{m} PR_k}{m} \quad k = 1,2,...,m \tag{45}$$

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.

8-3 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 (Mohammed & Rashid, 2023). 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 (BestFitness) و بهترین موقعیت (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 قبلی ($fittness_i$) در طول تکرارها بدست می‌آیند. با استفاده از یک متغیر تصادفی به نام r، احتمال 50% 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 $[0,1]$ است. بنابراین، اگر عدد تصادفی p بزرگتر از 0.18 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 0 تا 1 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

[1] Fox-inspired Optimization Algorithm (FOX)

شدن صدا $Time_S_T_{it}$، به دست می‌آید.

$$Dist_S_T_{it} = Sp_S \times Jump \quad (46)$$

که در آن سرعت صوت در محیط Sp_S برابر با 343 در هوا و $Time_S_T_{it}$ یک عدد تصادفی در محدوده [0، 1] است. it تعداد تکرارها است که از 1 تا 500 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.

$$Sp_S = \frac{BestPosition_{it}}{Time_S_T_{it}} \quad (47)$$

برای یافتن مسافت پیموده شده صوت از رابطه (46) 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 2 تقسیم می شود زیرا فاصله سنسور و جسم نصف فاصله‌ای است که توسط موج صوتی پیدا شده است.

$$Dist_Fox_\mathrm{Prey}_{it} = Dist_S_T_{it} \times 0.5 \quad (48)$$

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.

$$Jump_{it} = 0.5 \times 9.81 \times t^2 \quad (49)$$

که در آن 9.81 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 0.18 باشد.

$$X_{(it+1)} = Dist_Fox_\mathrm{Prey}_{it} \times Jump_{it} \times c_1 \quad (50)$$

اما اگر مقدار p کمتر از 0.18 باشد، موقعیت جدید با رابطه زیر محاسبه می‌شود.

$$X_{(it+1)} = Dist_Fox_\mathrm{Prey}_{it} \times Jump_{it} \times c_2 \quad (51)$$

مقدار c_1 و c_2 به ترتیب 0.18 و 0.82 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 0.18 باشد، به این معنی است که روباه قرمز به سمت شمال شرقی می‌پرد. در نتیجه، برای یافتن موقعیت جدید، هم $Dist_Fox_\mathrm{Prey}_{it}$ 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 0.18 باشد، این بدان معنی است که شانس کشتن طعمه کم است (18٪). بنابراین، هر دو $Dist_Fox_\mathrm{Prey}_{it}$ 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 (7) و (8) محاسبه $MinT$و متغیرها را نشان می‌دهد. $MinT$با یافتن حداقل tt محاسبه می‌شود.

$$tt = \frac{sum\left(Time_{ST_{it}}(i,:)\right)}{\text{dimension}},\ \ MinT = Min(tt) \tag{52}$$

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

$$a = 2\times\left(it - \left(\frac{1}{Max_{it}}\right)\right) \tag{53}$$

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 $X_{(it+1)}$.

$$X_{(it+1)} = BestX_{it}\times rand(1,\text{dimension})\times MinT\times a \tag{54}$$

4 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 (10روزه) و بلندمدت (30 روزه) با استفاده از روش یادگیری ماشین به نام شبکه عصبی مصنوعی[2] بهینه سازی شده توسط شش الگوریتم بهینه سازی فرابتکاری 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:

[1] Bitcoin
[2] Artificial Neural Network (ANN)

1-4 پیش پردازش داده‌ها

مرحله آماده‌سازی داده‌ها قبل از پردازش را، پیش‌پردازش[1] 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[2]، داده‌های پرت[3]، نرمال‌سازی داده‌ها[4] و تقسیم‌بندی داده‌ها[5] می باشد.

1-1-4 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[6] 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.

2-1-4 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[7]» مورد استفاده را دچار مشکل کنند. بنابراین بایستی تا حد ممکن، تاثیر داده‌های پرت را کاهش داد.

3-1-4 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[8]، نرمال‌سازی نمره زد[9]، مقیاس گذاری اعشاری[10] و ... اشاره کرد. به منظور کاهش اثرات مقیاس

---

[1] Preprocess
[2] Missing Data
[3] Outliers
[4] Normalization
[5] Data Splitting
[6] Noise
[7] Classifier
[8] Min-Max
[9] Z-Score
[10] Decimal Scaling

داده‌ها بر روی نتایج مدل، نرمال سازی داده‌ها در بازه 0 تا 1 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 (55) 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 0 و 1 می‌باشند.

$$XN = a + \frac{X - X_{\min}}{X_{\max} - X_{\min}} \times (b - a) \qquad (55)$$

4-1-4 تقسیم‌بندی داده‌ها

در یادگیری ماشین معمولا داده‌ها را به دو قسمت تفکیک می‌کنند. مجموعه‌داده‌های آموزش و آزمایش. در این تحقیق از 80 درصد از ابتدای مجموعه داده‌های قیمت بیت کوین به عنوان داده‌های آموزش و 20 درصد باقی‌مانده از انتهای داده‌ها، به عنوان داده‌های آزمایش استفاده شده است.

داده‌های آموزشی (Training set): از این بخش از داده‌ها به منظور ایجاد و آموزش مدل‌ها و الگوریتم‌های مختلف یادگیری ماشین و برآورد پارامترهای آن استفاده می‌شود.

داده‌های آزمایشی (Test set): 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.

2-4 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، می‌تولند 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 دخللت خارجی توسط الگوریتم‌های فراابتکاری جایگزین می‌شود. 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.

---

[1] Weight
[2] Bias

روند کار بدین صورت است که در ابتدا شبکه عصبی مورد نظر را تعریف می‌کنیم. ساختار این 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 (مینیمم سازی) 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: 1- خطای مینیمم مربعات برای شبکه از یک آستانه کمتر باشد 2- 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 بهینه سازی 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 (56) 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.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z}_i)} \qquad (56)$$

در این پژوهش سعی داریم تا به پیش‌بینی قیمت رمز ارز بیت کوین در بازه‌های زمانی کوتاه‌مدت (10روزه) و بلندمدت (30 روزه)، با استفاده از روش شبکه عصبی مصنوعی بهینه سازی شده توسط الگوریتم‌های (PO)،(HBO)،(SPO)، (GPC)، (FHO) و (FOX) بپردازیم. 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[1]، قیمت بالایی رمز ارز[2]، قیمت پایینی رمز ارز[3]، حجم معاملات[4] و ارزش بازار[5] 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 1، 2، 3، 4 و 5 روز قبل یعنی $(X-1, X-2, X-3, X-4, X-5)$ است، به عبارتی میزان تاخیرها را 1، 2، 3، 4 و 5 روز قبل در نظر گرفته‌ایم و متغیر خروجی میزان متغیر قیمت نهایی رمز ارز[6] در 10 روز بعد یعنی $X+10$ است که قصد داریم آن را پیش‌بینی کنیم. به عبارت دیگر میزان متغیر قیمت رمز ارز در 10 روز آینده را تابعی از میزان متغیرهای ورودی در 1، 2، 3، 4

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

و 5 روز قبل در نظر گرفته‌ایم که این عبارت در رابطه زیر نمایش داده شـده اسـت. با توجه به اینکه 5 متغیر ورودی هر کدام 5 تاخیر بر روی آن‌ها لحاظ شده‌اند، لذا در پژوهش حاضر 25 متغیر ورودی و 1 متغیر خروجی داریم.

$$X+10 = f(X-1,\ X-2,\ X-3,\ X-4,\ X-5) \qquad (57)$$

همچنین یک بار نیز متغیر خروجی میزان متغیر قیمت نهایی رمز ارز در 30 روز بعد در نظر گرفته شـده اسـت، یعنی $X+30$ که قصـد داریم آن را پیش‌بینی کنیم. به عبارت دیگر میزان متغیر قیمت رمز ارز در 30 روز آینده را تابعی از میزان متغیرهای ورودی در 1، 2، 3، 4 و 5 روز قبل در نظر گرفته‌ایم که این عبارت در رابطه زیر نمایش داده شده است.

$$X+30 = f(X-1,\ X-2,\ X-3,\ X-4,\ X-5) \qquad (58)$$

3-4 معیارهای ارزیابی و اعتبارسنجی

در این پژوهش، به منظور ارزیابی و مقایسـه کارآیی الگوریتم‌های بهینه سـازی، بهینه‌سـاز سیاسی (PO)، بهینه‌ساز مبتنی بر سلسله مراتب رتبه در شرکت (HBO)، بهینه‌ساز رنگ‌های تصـادفی (SPO)، الگوریتم سـاخت اهرام جیزه (GPC)، بهینه‌سـاز شـاهین آتشـین (FHO) و الگوریتم بهینه‌سازی فاکس (FOX) در بهینه سازی شبکه عصبی مصنوعی و پیش‌بینی قیمت رمز ارز، از معیارهای معتبر بـه شـرح موارد زیر اسـتفاده شـده اسـت. در این روابط، $x_{imeas}, x_{ipred}, \overline{x}, n$ به ترتیب برابر با تعداد متغیر اندازه‌گیری شده، مقدار میانگین متغیر اندازه‌گیری شده، مقدار متغیر پیش‌بینی شده و مقدار متغیر اندازه‌گیری شده می‌باشد.

1-3-4 ضریب تعیین

ضـریب تعیین[1] که با R^2 نمایش داده می‌شـود، قدرت توضـیح دهندگی مدل را نشـان می‌دهد. ضریب تعیین نشان می‌دهد که چند درصد از تغییرات متغیر وابسته توسط متغیرهای مستقل توضیح داده می‌شود. ضریب تبیین بیان کننـده بخشی از کل واریانس مقادیر مشـاهده ای اسـت کـه توسـط مقادیر شبیه سازی شده توجیه می شود. ضریب تبیین بین صفر تا یک تغییر می کند و مقـدار بهینهٔ آن یک است و این حالت مربوط به زمانی اسـت کـه در آن مقادیر شبیه سازی شـده دقیقا مشابه مقادیر مشاهداتی باشد.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_{imeas} - x_{ipred})^2}{\sum_{i=1}^{n}(x_{imeas} - \overline{x})^2} \qquad (59)$$

[1] Determination Coefficient

2-3-4 میانگین مربعات خطا

میانگین مربعات خطا[1] (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:

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} [(x_{imeas} - x_{ipred})^2] \qquad (60)$$

3-3-4 مجذور میانگین مربعات خطا

مجذور میانگین مربعات خطا[2] (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.

$$RMSE = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} [(x_{imeas} - x_{ipred})^2]} \qquad (61)$$

4-3-4 میانگین قدر مطلق خطا

میانگین قدر مطلق خطا[3] (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 است و مقدار بهینهٔ آن صفر است.

$$MAE = \frac{1}{n} \times \sum_{i=1}^{n} | x_{imeas} - x_{ipred} | \qquad (62)$$

5-3-4 مربع خطای نسبی

مربع خطای نسبی[4] (RSE) نشان‌دهنده‌ی عدم تناسب مدل با داده های موجود است. به عبارت ساده تر، 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.

---

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

$$RSE = \frac{\sum_{i=1}^{n}(x_{ipred} - x_{imeas})^2}{\sum_{i=1}^{n}(\bar{x}_{meas} - x_{imeas})^2} \quad (63)$$

6-3-4 واریانس تبیین شده

در آمار، واریانس تبیین شـــده[1] (EVS)، تناســـبی را که یک مدل ریاضـــی برای تغییرات (پراکندگی) یک مجموعه داده معین انجام می‌دهد، اندازه گیری می کند. بهترین مقدار ممکن برای این معیار 1 است ، مقادیر پایین بدتر هستند.

$$EVS = 1 - \frac{Var\{x_{imeas} - x_{ipred}\}}{Var\{x_{imeas}\}} \quad (64)$$

4-4 پیش بینی قیمت رمز ارز بیت کوین در بازه زمانی کوتاه مدت 10 روز آینده

پیش‌بینی قیمت رمز ارز توسط شبکه عصبی مصنوعی بهینه سازی شده توسط الگوریتم‌های بهینه ســـازی PO، HBO، SPO، GPC، FHO و FOX در بازه‌ی زمانی کوتاه‌مدت 10روزه، در نرم افزار MATLAB انجام شـــد که نتایج و معیارهای ارزیابی در دو حالت آموزش و آزمایش به شرح زیر می‌باشند.

جدول 2: نتایج پیش‌بینی قیمت رمز ارز بیت کوین در بازه زمانی کوتاه مدت 10 روز آینده

پارامتر	مرحله	R^2	RMSE	MSE	MAE	RSE	EVS
ANN-PO	Train	0/9561	0/0006215	0/02493	0/0096597	0/043963	0/79038
	Test	0/83079	0/011526	0/10736	0/092768	0/34485	0/58468
ANN-HBO	Train	0/9421	0/00099932	0/031612	0/017222	0/070688	0/74085
	Test	0/80913	0/019678	0/14028	0/11629	0/58872	0/53182
ANN-SPO	Train	0/973	0/00038178	0/019539	0/0074159	0/027005	0/83567
	Test	0/86383	0/0095864	0/09791	0/082818	0/28681	0/62968
ANN-GPC	Train	0/95972	0/00084298	0/029034	0/0090983	0/059629	0/75808
	Test	0/88095	0/0094913	0/097423	0/066314	0/28396	0/5484
ANN-FHO	Train	0/9622	0/00066087	0/025707	0/010619	0/046748	0/78473
	Test	0/84125	0/0057626	0/075912	0/056109	0/17241	0/58619
ANN-FOX	Train	0/95246	0/00067399	0/025961	0/010702	0/047676	0/78174
	Test	0/79594	0/01177	0/10849	0/094447	0/35214	0/52999

5-4 پیش بینی قیمت رمز ارز بیت کوین در بازه زمانی بلند مدت 30 روز آینده

پیش‌بینی قیمت توســـط الگوریتم‌های بهینه ســـازی PO، HBO، SPO، GPC، FHO و

[1] Explained Variance Score

FOX در بازه‌ی زمانی بلندمدت 30 روزه انجام شد که نتایج و معیارهای ارزیابی در دو حالت آموزش و آزمایش به شرح زیر می‌باشند.

جدول 3: نتایج پیش‌بینی قیمت رمز ارز بیت کوین در بازه زمانی بلند مدت 30 روز آینده

پارامتر	مرحله	R^2	RMSE	MSE	MAE	RSE	EVS
ANN-PO	Train	0/91066	0/0013328	0/036507	0/01892	0/089877	0/70024
	Test	0/70787	0/010778	0/10382	0/08311	0/32637	0/45279
ANN-HBO	Train	0/91526	0/0013565	0/036831	0/014952	0/091478	0/69766
	Test	0/66278	0/011313	0/10636	0/080647	0/34256	0/41925
ANN-SPO	Train	0/94889	0/00075849	0/027541	0/012901	0/05115	0/77385
	Test	0/69917	0/031993	0/17887	0/1525	0/96876	0/41425
ANN-GPC	Train	0/93156	0/0010303	0/032098	0/014765	0/069477	0/73684
	Test	0/72114	0/016069	0/12676	0/10337	0/48657	0/45455
ANN-FHO	Train	0/9296	0/0010925	0/033053	0/020307	0/073673	0/73388
	Test	0/69885	0/015261	0/12353	0/095192	0/4621	0/40536
ANN-FOX	Train	0/94868	0/0007825	0/027973	0/015375	0/052769	0/77289
	Test	0/70728	0/024249	0/15572	0/12464	0/73428	0/38109

5 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 10 روزه و بلندمدت 30 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 0 تا 1 می‌پردازیم زیرا وارد کردن داده‌ها به صورت خام باعث کاهش سرعت و دقت مدل می‌شود. سپس با تقسیم بندی داده‌ها از 80 درصد از ابتدای مجموعه داده‌های قیمت بیت کوین به عنوان داده‌های آموزش و 20 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 حللت آموزش و آزمایش می‌پردازیم. معیارهای ارزیابی و اعتبارسنجی مدل‌ها شامل (R^2)، (RMSE)، (MSE)، (MAE)، (RSE) و (EVS) 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 10 روزه و بلند مدت 30 روزه، عملکرد پیش بینی شبکه عصبی بهینه‌سازی شده توسط الگوریتم GPC به ترتیب با ضریب تعیین 0/88 و 0/72 بهتر از سایر الگوریتم‌ها بوده است.

6 تعارض منافع

هیچگونه تعارض منافع توسط نویسندگان بیان نشده است.

7 منابع

Askari, Q., Saeed, M., & Younas, I. J. E. S. w. A. (2020). Heap-based optimizer inspired by corporate rank hierarchy for global optimization, *161*, 113702 .

Azizi, M., Talatahari, S., & Gandomi, A. H. J. A. I. R. (2023). Fire Hawk Optimizer: A novel metaheuristic algorithm, *56*(1), 287-363 .

Chhajer, P., Shah, M., & Kshirsagar, A. J. D. A. J. (2022). The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction, *2*, 100015 .

Choithani, T., Chowdhury, A., Patel, S., Patel, P., Patel, D., & Shah, M. J. A. o. D. S. (2024). A comprehensive study of artificial intelligence and cybersecurity on bitcoin, crypto currency and banking system, *11*(1), 103-135 .

Dutta, A., Kumar, S., Basu, M. J. J. o. r., & management, f. (2020). A gated recurrent unit approach to bitcoin price prediction, *13*(2), 23 .

Harifi, S., Mohammadzadeh, J., Khalilian, M., & Ebrahimnejad, S. J. E. I. (2021). Giza Pyramids Construction: an ancient-inspired metaheuristic algorithm for optimization, *14*(4), 1743-1761 .

Hornik, K., Stinchcombe, M., & White, H. J. N. n. (1989). Multilayer

feedforward networks are universal approximators, *2*(5), 359-366 .
Hussein, N., & Abdulazeez, A. M. J. I. J. o. C. S. (2024). Bitcoin Price Prediction Using Hybrid LSTM-GRU Models, *13* .(1)
Kaveh, A., Talatahari, S., & Khodadadi, N. J. E. w. C. (2022). Stochastic paint optimizer: theory and application in civil engineering, 1-32 .
Koker, T. E., Koutmos, D. J. J. o. R., & Management, F. (2020). Cryptocurrency trading using machine learning, *13*(8), 178 .
Mohammed, H., & Rashid, T. J. A. I. (2023). FOX: a FOX-inspired optimization algorithm, *53*(1), 1030-1050 .
Smith, M. (1993). *Neural networks for statistical modeling*: Thomson Learning .
skeanh saada, n., 'ela asma'eal, z., & mhmdrda, r. (1402). ara'eh mdl peash bana bazdha bat kewan ba astfadh az rwsh habrada aadguara 'emaq - alguwratm tjzah sagunal (CEEMD- DL). aqtsad mala, 17(62), 217-238 .

mhmdbaqr, m. (1381). mbana shbkehhaa 'esba hwsh mhasbata (Vol. 0): danshguah sn'eta amarkebar (pelatkenake thran), wahdtfrsh, mrkez nshr perwfswr.

Ayyildiz, N. and O.J.H.(2024). Iskenderoglu, How effective is machine learning in stock market predictions?
Jain, S., et al.(2024). Exploring the Future of Stock Market Prediction through Machine Learning: An Extensive Review and Outlook, 12(4): p. 1-10.

Ajiga, D.I., et al.(2024). Machine learning for stock market forecasting: a review of models and accuracy, 6(2): p. 112-124.

Phuoc ,T., et al.(2024). Applying machine learning algorithms to predict the stock price trend in the stock market–The case of Vietnam, 11(1): p. 1