A Comparative Analysis of Ensemble–Metaheuristic Algorithms for Copper Price Forecasting: The NGO-AdaBoost Hybrid Approach

Xianming Liu
School of Economics and Management, Hubei Polytechnic University, Huangshi 435000, China
E-mail: Liu.xm@126.com

Keywords: copper price forecasting, global price of copper, adaboost, northern goshawk optimization, artificial intelligence (AI)

Received: March 3, 2025

On international trade marketplaces, copper prices fluctuate a lot. Since copper is a valuable material, changes in its price may have an impact on certain nations' economies' ability to grow sustainably. The price of copper is a major issue for investors, policymakers, and futures traders. Governments and businesses that rely on copper mining must be able to forecast copper prices to make critical choices. This price is predicted in this article using an artificial intelligence technique. This paper proposes a hybrid forecasting model using the Adaptive Boosting model and the Northern Goshawk Optimization algorithm to enhance copper price predictions in terms of accuracy and reliability. A hybrid NGO-AdaBoost model was created to benefit from both the efficacy of ensemble learning and the exploratory potential of metaheuristic optimization. For this reason, time series datasets of copper prices from January 2014 to October 2023 were created using historical data, including open, high, low, and close prices and volume. Key metrics were used to assess the predictive accuracy of the model, and the results demonstrated high predictive accuracy with an R-squared (R^2) value of 0.9919, a Mean Absolute Percentage Error (MAPE) of 0.6733, and a Mean Absolute Error (MAE) of 0.0268 on the test dataset. Additionally, comparative analysis shows that the suggested NGO-AdaBoost model outperforms other benchmark techniques and traditional forecasting models like Autoregressive Integrated Moving Average and Support Vector Regression in terms of prediction, accuracy, and stability. This confirms the model's capacity to capture the intricate dynamics of copper price volatility. The results demonstrated that the proposed hybrid scheme could more reliably and accurately forecast the price of copper than the other schemes used in this study. The research presented in this paper provides a reliable source for predicting future changes in copper prices.

Povzetek: Podan je pregled in nato opisan izviren hibridni NGO-AdaBoost model za napoved cene bakra, ki združuje metahevristično optimizacijo in ansambelsko učenje ter dosega bolj kvalitetno napovedovanje kot ARIMA, SVR in LSTM.

1 Introduction

Copper is a valuable metal for many industries and plays a significant role in many financial sector enterprises. Copper is the third in the global consumed metal after iron and aluminum because of its malleability and conductivity. For evaluating the cost, availability, and demand of copper, the futures markets of renowned futures trading exchanges serve as the principal trading platforms, like LME (London Metal Exchange), COMEX (New York Commodity Exchange), and SHFE (Shanghai Futures Exchange) [1,2]. It is important to mention that these factors are greatly impacted by investment flows and currency exchange rates [3]. The financial success of several sectors is directly contingent upon the fluctuation in metal prices, including the price of copper. Copper production exerts a significant reliance on developing economies, influencing their trajectory both domestically and strategically [4]. Price fluctuations for copper have a big effect on the fiscal or financial income of a nation or a corporation, making it a crucial aspect to consider. The cost of copper is predominantly influenced by the efficiency of copper providers in extracting as well as transporting the metal, together with the demand for products and services that rely on copper. Regulators, investors, and companies that manufacture copper must all be able to predict supply and demand shocks and market patterns with precision [5]. Accurate pricing projections enable policymakers to manage the market with more effectiveness. Copper producers can enhance the efficiency and responsiveness of their production processes. As a result, investors may more effectively develop lucrative investing plans for both the short and long term. One of the biggest obstacles to successful forecasting is determining the precise traits of a time series [6]. Because financial time series are nonlinear, volatile, and regime-shifting, forecasting copper prices remains extremely challenging despite its significance. Particularly in times of market turbulence, traditional statistical models frequently fail to capture these intricate dynamics, producing forecasts that are unstable or erroneous. Numerous econometric schemes, including wavelet-ARIMA [7], generalized autoregressive conditional heteroscedastic (GARCH) schemes [8,9], as well as ARIMA (autoregressive integrated moving average) [3,8], have been created to predict copper prices. These time series model; however, are unable to adequately represent the time series' nonlinear nature. As a result, advances in computational intelligence, machine learning (ML)[10][11][12], and AI tactics have been thoroughly researched and employed to determine and document diverse traits of the investigated time series of copper prices [1–4,8].

To address the issue of modeling the nonlinear and unstable nature of copper prices, this study proposes a hybrid artificial intelligence (AI) model that integrates ensemble learning and metaheuristic optimization. The main goal is to create a strong model that performs better than traditional machine learning techniques and can better represent the dynamic behavior of copper prices. This study provides the assumption that by utilizing the global optimization power of the Northern Goshawk Optimization algorithm and the adaptive learning capability of AdaBoost, the suggested NGO-AdaBoost hybrid model will outperform conventional models in forecasting accuracy. Metaheuristic algorithms are generic search strategies used to efficiently explore big and complicated solution spaces, which is thereby extremely appropriate for optimization issues. To this extent, an NGO is employed to optimize the individual base learners within the AdaBoost algorithm to achieve maximum accuracy and stability against evolving market conditions. This study helps bridge the gap by presenting a more precise and reliable approach for predicting copper prices, serving as a helpful tool for investors, policymakers, and futures traders to analyze upcoming changes in copper prices and make well-informed choices.

Ensemble classifiers are a popular AI technique for data categorization. Many weak classifiers, or those with a lower classification accuracy, are utilized in the ensemble classifiers approach. Every classifier has a weight that influences the classification outcome. The process of applying these weights to determine the categorization outcome is known as "weight voting." One of the most popular ensemble classifier tactics that takes advantage of the weight-voting process is AdaBoost. The proposal was made by Freund and Schapire in 1997 [13]. The AdaBoost technique may improve classification accuracy and is simple to use. It may be used for many classifiers as well.

The Northern Goshawk Optimization (NGO) offers highly desirable quasi-optimal solutions for optimization problems. It excels in addressing real-world scenarios and exhibits impressive performance in solving optimization issues. Moreover, it outperforms comparable algorithms by effectively balancing the exploration and exploitation processes to identify ideal resolutions. Fewer input parameters are needed for the recommended optimizer. The northern goshawk is the sole member of the Accipiter genus that preys on a diverse range of animals [14]. Males in this species are somewhat more widespread than females, and they are spread in both Eurasia and North America. The northern goshawk divides its hunting process into two stages: the first involves it moving swiftly in the direction of its target when it spots it, and the second involves a brief tail chase to catch up with the victim [15]. This article examines the NGO algorithm's capacity to predict the price of copper as a consequence. This work's contribution is broken down as follows:

- Acknowledged the influence that copper price volatility has on economies and the international trade markets, emphasizing the significance of precise copper price prediction for investors, policymakers, and futures traders.
- This study contributes to the field by showcasing the potential of metaheuristic optimization in ensemble learning models and proposing the first integration of an NGO with AdaBoost for copper price forecasting. Furthermore, it offers a comprehensive examination of multiple metaheuristic optimization algorithms concerning the prediction of copper prices, offering valuable insights into how different optimization techniques impact forecasting accuracy. By emphasizing the benefits of incorporating advanced optimization algorithms, this study paves the way for more reliable and efficient predictive models in commodity price forecasting. The study also contributes to the literature by demonstrating how hyperparameter tuning enhances model performance and provides a framework for additional research in this field.
- Nine years and 10 months' worth of copper price time series datasets, including volume and the open, high, low, and close values, were gathered from historical data.
- Developed the NGO-AdaBoost model and showed that, with a coefficient of determination (R^2) of 0.9919, it performed better in forecasting copper prices when compared to other ML tactics.

This article is organized in the following manner: A comprehensive examination of the data source and all of its components is provided in Section 2. The scheme, optimizers, and other elements utilized in this paper are analyzed in the third part. In the fourth and fifth parts, the outcomes of the schemes utilized in this article are compared. After evaluating the chosen schemes in this article, the conclusion is covered in the next section, and the last section discusses potential future projects that might build on the scheme employed in this article.

2 Literature review

Esperanza García-Gonzalo et al. [16] concentrated on predicting copper spot prices from COMEX by using support vector regression (SVR) and several model architectures. The accuracy of three-time series analyses was examined, and the hybrid direct-recursive approach emerged as the most accurate in terms of numerical outputs. Jiahao Chen et al. [17] utilized an LSTM (Long Short-Term Memory) AI model to make forecasts about the pricing of copper. The effectiveness of the LSTM model was improved by using a simulated annealing approach to tune hyperparameters. Relationship assessment was utilized to address feature engineering. The scheme used economic factors that have a strong correlation with copper prices, such as the West Texas Intermediate Oil Price, Gold Price, and Silver Price.

In a study by Gabriel Astudillo et al. [18], SVR was specifically used to predict the closing prices of copper at LME over multiple periods, with grid search and balanced cross-validation being used to select the best model for each forecast timeframe. The experimental results demonstrate that the parameters of SVR remain constant over prediction intervals and the prior values used for estimations.

A novel forecasting model that blends interval and point forecasting is presented by Yifei Zhao et al. [19]. The interval forecasting model was validated by gathering the interval forecasting results and using the copper and aluminum pricing data. The evaluation results show that the accuracy of the LSTM model is 0.099021.

Hongyuan Luo et al. [20] introduce a novel multi-step-ahead forecasting model of copper price with an error correction scheme-based and genetic algorithm-enhanced long short-term memory (GA-LSTM) structure. The predictive efficacy of the introduced framework is supported by a 30-year real copper price time series. A 30-year real iron ore price time series is also utilized to show the robustness and generalization capability of the introduced design. The outcomes demonstrated that the F-TS-GA-LSTM-EC hybrid model was effective and more accurate in copper price prediction.

Table 1: Summary of literature review on copper price forecasting.

Authors	Method	Dataset Description	Performance Metrics	Year
García-Gonzalo et al. [16]	SVR (Support Vector Regression)	COMEX monthly copper spot prices (1960-2019)	MAE: 144.21 RMSE: 170.15 MAPE: 2.36%	2023
Jiahao Chen et al. [17]	LSTM + Simulated Annealing	Daily copper + macroeconomic indicators (1990-2009), from Investing.com	MSE: 0.000569 (test set)	2023
Gabriel Astudillo et al. [18]	SVR + Grid Search + Cross-Validation	Daily copper prices from LME (2006-2018)	RMSE: 0.0177 (5-day) R = 0.9582	2020
Yifei Zhao et al. [19]	VMD-SSA-LSTM + Interval Forecasting	Daily copper and aluminum prices (2012-2022), Shanghai Futures Exchange	R^2 = 0.99827 RMSE = 763.01 MAE = 543.20 MAPE = 0.8789%	2023
Hongyuan Luo et al. [20]	F-TS-GA-LSTM-EC (Factor + Time Series + GA-LSTM + Error Correction)	Monthly copper prices + 8 factors (1991-2021), 360 records, IndexMundi	R^2 = 0.920 MAE = 257.26 RMSE = 330.04 MAPE = 4.03%	2022
Hasel Amini Khoshalan et al. [21]	GEP, ANN, ANFIS, ANFIS-ACO	Monthly copper + 8 influencing factors (1990-2020), IndexMundi	R^2= 0.981 RMSE = 356.51 MAE = 239.11	2021

Note: SVR=Support Vector Regression, LSTM= Long Short-Term Memory, SA= Simulated Annealing, VMD= Variational Mode Decomposition, SSA= Sparrow Search Algorithm, GA= Genetic Algorithm, F-TS-GA-LSTM-EC= Factor + Time Series + Genetic Algorithm-optimized LSTM + Error Correction, GEP= Gene Expression Programming, ANN= Artificial Neural Network, ANFIS= Adaptive Neuro-Fuzzy Inference System, ACO= Ant Colony Optimization, R^2= Coefficient of Determination, MAE= Mean Absolute Error, RMSE= Root Mean Square Error, MAPE= Mean Absolute Percentage Error, COMEX= Commodity Exchange (New York), LME= London Metal Exchange.

Statistical models such as ARIMA and GARCH, while simple, often do not capture the non-stationary and non-linear commodity price dynamics. They are incapable of responding well to the incidence of structural breaks and abrupt changes that are common in metals markets. Machine learning models such as SVR and standalone LSTM have done better than statistical models but still fail to capture the short- and long-term dependencies simultaneously. SVR, in particular, does not produce consistent performance for varying forecast horizons due to its sensitivity to kernels. More sophisticated hybrids, such as GA-LSTM and ANFIS-ACO, have high improvement, but they can be vulnerable to parameter tuning sensitivity, local optima, or low robustness to unforeseen volatility patterns. These drawbacks motivated the design of the NGO-AdaBoost hybrid model, which combines.

- The discoverability of NGOs to correct model parameters and break out of local minima
- AdaBoost's adaptive ensemble learning capability to strengthen weak learners and counteract overfitting, especially in noisy or unstable data conditions

- A strong design tested with a 9 years and 10 months' time series, ensuring its durability, accuracy, and reactivity to complex patterns in copper price dynamics.

3 Dataset

This research examines the prediction of copper prices. From January 2014 to October 2023, pricing data was collected for the input parameters reviewed and changes in copper prices. The prices in this article are expressed in US dollars per pound (in USD/LBS). Various variables may affect copper price fluctuations. The variables used in this article are volume and open, high, low, and close prices.

3.1 Data Pre-processing

Historical copper prices based on global trends were obtained from Yahoo Finance and TradingView. Data cleaning was done intensively to handle any possible missing or inconsistent values in the data to ensure that the data input into the machine learning models is accurate and clean. Data cleaning involves the detection and fixing of errors, inconsistencies, or missing values in the dataset so that data input into the machine learning models is clean and accurate. Because missing or erroneous data could lead to faulty predictions and stop the model from functioning, this process is imperative. Following data cleaning, feature normalization was conducted so that all input features, like open, high, low, volume, and close prices, were normalized and values were between 0 and 1. Standardizing features enhances machine learning models because it allows them to learn from features of different scales. To test the model, the data was split into training and test sets, where 80% was used for training and 20% for testing. This split enables the generalization of the model to unseen data to be tested by preventing overfitting and ensuring that the performance of the model can be properly measured. This 80/20 ratio is common in machine learning to achieve a balance between training the model sufficiently and testing it on a big enough sample of unseen data.

3.2 Historical data

The opening price is the price of a share at the start of trading and is a reliable indicator of the daily volatility of the financial market. Because the financial market is like an auction where buyers and sellers compete to find the highest bidder, the opening price does not necessarily have to match the closing price of the day before.

The previous day's highest and lowest prices are captured, which gives data about the average volatility of the market on a trading day and how it impacts the final closing price. The adjusted closing price is the final price of a property after adjustment for any dividends or corporate activity that has happened before the opening of the subsequent market day.

Adjusted closing price is generally employed for analyzing the current returns or carrying out a detailed analysis of past returns.

Volume is the total number of contracts or shares that are transferred in a security or the entire market within a definite time frame.

To build an efficient input space for prediction, the five important features of the copper price data—open, high, low, close, and volume (OHLCV)—were utilized. These are widely employed in financial time series analysis because both indicate the direction of price as well as activity in the market during a period. Before training, all variables were normalized using min-max scaling to achieve consistency across different ranges. With all five variables, the highest prediction and stability among the models under test were obtained, confirming their individual and combined roles in the forecasting model.

3.3 Statistical values

The dataset is thoroughly examined in the report, which is shown in Table 2. A comprehensive statistical representation of the data pertaining to the input and output attributes is given in the following table. This process guarantees that all of the information can be understood completely. The table contains a large number of statistical measures, including the mean, variance, count, 25%, 50%, 75%, minimum (min), maximum (max), and standard deviation (Std.). By using these measurements, a thorough and accurate analysis of the data can be accomplished.

Table 2: A statistical summary of the dataset

	count	mean	std.	min	25%	50%	75%	max	variance
Open price	2467	3.103505	0.699136	1.9385	2.617	2.95	3.5855	4.92	0.488791
High price	2467	3.132707	0.707937	1.98	2.64175	2.977	3.63	5.0395	0.501175
Low price	2467	3.073633	0.690216	1.9355	2.5975	2.9225	3.556	4.81	0.476399
Volume	2467	31651.08	38277.74	10	290	1500	61380	230070	1.47E+09
Close price	2467	3.103989	0.6999	1.9435	2.62025	2.951	3.58725	4.9375	0.48986

4 Methodology

This work presents the design of an FOA-AdaBoost, SSA-AdaBoost, and NGO-AdaBoost model for predicting copper prices. These schemes are described in a general sense in this section. The AdaBoost model and the FOA, SSA, and NGO algorithms are initially proposed as follows:

4.1 Adaptive boosting

One of the most powerful recognition strategies is AdaBoost, which blends many weak predictors to create an effective predictor [22] [23] . AdaBoost training, by Lu et al. [24], has greater dispersion weights in the case of greater errors and smaller dispersion weights when errors are lesser. To improve the expected outcome, the samples are subsequently trained using the revised weight dispersion [25]. The following are the AdaBoost computation stages [26] [27].

$$U = \{(x_i, y_i)\}_{i=1}^{N}, y_i \in \{-1, +1\} \tag{1}$$

Step 1: Initialize weights

Set sample weights as uniform:

$$D_1(i) = \frac{1}{N}, \text{ for } i = 1,2, \dots, N \tag{2}$$

Step 2: Train weak learner error:

$$\varepsilon_t = \sum_{i=1}^{N} D_t(i) \cdot \mathbb{I}(f_t(x_i) \neq y_i) \tag{3}$$

Step 3: Calculate learner weight

The importance weight of the learner is computed as:

$$\alpha_t = \frac{1}{2} \log \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \tag{4}$$

Step 4: Update sample weights

Update the weight distribution:

$$D_{t+1}(i) = \frac{D_t(i) \cdot e^{-\alpha_t y_i f_t(x_i)}}{Z_t} \tag{5}$$

where Z_t is a normalization factor ensuring that $\sum_i D_{t+1}(i) = 1$.

Step 5: Final hypothesis

Following T rounds, the terminal strong classifier consists of a weighted ensemble of the weak learners:

$$F(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t f_t(x) \right) \tag{6}$$

Loss Function:

AdaBoost attempts to minimize the exponential loss function over the training data:

$$\mathcal{L}(F) = \sum_{i=1}^{N} e^{-y_i F(x_i)} \tag{7}$$

This loss function is convex and imposes a larger penalty on misclassified samples, forcing the model to focus on hard-to-classify points.

The steps above define AdaBoost's internal training logic, but the choice of hyperparameters such as n_estimators, learning_rate, and random_state significantly affects the algorithm's performance. Before the boosting process begins, this research optimizes these parameters using the NGO algorithm. NGO is employed as a pre-training hyperparameter tuning technique to ascertain the ideal configuration of AdaBoost's hyperparameters through a two-phase bio-inspired search strategy (local exploitation and global exploration). This optimization lowers validation loss while increasing the final model's accuracy and generalization. Therefore, without altering AdaBoost's internal processes, the NGO strategically improves its conditions.

4.2 Metaheuristic optimization of adaboost hyperparameters

Five metaheuristic algorithms were utilized to fine-tune the most important hyperparameters of the AdaBoost model to maximize its predictive performance: GA, PSO, FOA, SSA, and NGO. These include the learning rate, number of estimators (n_estimators), and random state initialization.

To minimize AdaBoost's exponential loss on a validation set, each metaheuristic investigated this search space. Table 3 displays the outcomes of the optimization procedure. Remarkably, the NGO algorithm produced the best overall performance across all evaluation metrics (R^2, MAPE, MAE, and RSE) with 800 estimators, a learning rate of 0.25, and a random state of 64. These results show that different algorithms converge on different parameter configurations. For example, GA and PSO favored higher estimator counts (1500 and 1200, respectively), while FOA and SSA selected somewhat lower values (700 and 1000). The higher learning rate (0.25) selected by NGO in comparison to GA and PSO (0.1) may have allowed for faster convergence and more responsive adaptation to data variations, especially in volatile financial time series like copper prices. The success of the NGO algorithm can be attributed to its two-phase hunting approach, which successfully balances global exploration (by identifying prey at random) with local exploitation (by pursuing and adjusting). This dual mechanism ensures more stable parameter tuning and helps avoid premature convergence, which is common in simpler algorithms like FOA, in contrast to SSA, which can lead to unpredictable updates in noisy datasets. Overall, this test demonstrates that an NGO is appropriate as the primary optimizer in the proposed framework for forecasting copper prices because it provides the most effective hyperparameter configuration for AdaBoost in addition to a dependable search procedure.

Table 3: AdaBoost hyperparameter optimization with five metaheuristic algorithms.

Ada Boost	Upper and lower band	GA	PSO	FOA	SSA	NGO
n_estimators	[100, 2000]	1500	1200	700	1000	800
learning_rate	[0.0001, 1]	0.1	0.1	0.2	0.2	0.25
random_state	[2, 100]	32	28	64	74	64

Take note that Figure 1 shows the pseudocode for integrating the AdaBoost model with Northern Goshawk Optimization (NGO). The figure demonstrates how the NGO optimizes the AdaBoost algorithm's primary hyperparameters, including the number of estimators, learning rate, and random state. AdaBoost performs better and is better able to handle the intricacies of the copper price prediction task thanks to this optimization process.

```
Inputs and initializations:
  - Input Training data (x_train, y_train)
  - Input Validation data (x_val, y_val)
  - Initialize searching boundaries for AdaBoost parameters
  - Initialize NGO parameters: population_size, max_iterations.

1. Initialize NGO:
  - Set individual's parameters as a vector.
  - Parameters: [n_estimators, learning_rate, max_depth]

2. Initial population assessment:
  For each individual:
    - Train AdaBoost model using the encoded parameters.
    - Compute fitness.

3. Do while i < max_iterations:
  a. For each individual:
    - Update current individual position:
      - Exploration and exploitation (Goshawk behavior).
    - Bound the found position to allowed boundaries.

  b. Evaluate New Individuals:
    - Build AdaBoost with new found hyperparameters.
    - Get new fitness value.

  c. Keep best results:
    - Keep best found individuals based on their fitness.
    - Update global best solution.

4. Return the best found individual as the optimal solution.

5. Train AdaBoost with optimal parameters found by NGO.
```

Figure 1: Pseudocode for the integration of northern goshawk optimization with adaboost.

4.3 Optimizer

4.3.1 Fruit fly optimization algorithm

The entrapment behavior exhibited by fruit flies served as the impetus for the formulation of FOA according to Fig. 2, a novel global optimization technique [28]. To determine the optimal solution to an issue with optimization, fruit flies are simulated by the FOA model. A specific flavor in the air is detected by employing sensitive olfactory organs, after which the target location is located by using the acute sight organs that have been retained. Vision, Osphresis foraging, population assessment, and initiation comprise the initial FOA.

The replies are first generated within a range defined by lower and upper borders. Here, $a_{x,y}$ displays the x-th resolution and the y subscript indicates the position of the element inside the x-th resolution, as defined by Equation (8).

$$a_{x,y} = \text{rand}\left(hb_y - lb_y\right) + lb_y \tag{8}$$

The lower limit is depicted as lb, whereas the upper bound is depicted as hb. The term "rand" refers to a haphazardly selected integer from a uniform distribution. Launching the population during the Osphresis foraging phase intensifies the alignment of each option, in which the solution is displaced a random distance from its existing position. The arithmetical expression is represented by Equation (9).

$$a_{x,y}^{(r+1)} = a_{x,y}^{(r)} \mp \text{rand } 0 \tag{9}$$

The current response is represented by $a_{x,y}^{(t)}$, while the fresh alignment is displayed by $a_{x,y}^{(t+1)}$. Additionally, a random value is selected from the range $O[2,2]$. The letter "t" displays the count of cycles. After the relocation, the fitness value is computed for each solution, and the avaricious selection procedure selects whether to keep the new position or the previous one. If the most recent response possesses a higher fitness value than the one before it, it will substitute the old answer. Alternatively, if the previous response remains within the population, the new solution will be declined. The algorithm halts and generates the best possible outcome when it reaches the specified termination condition. Fig. 3 illustrates the method.

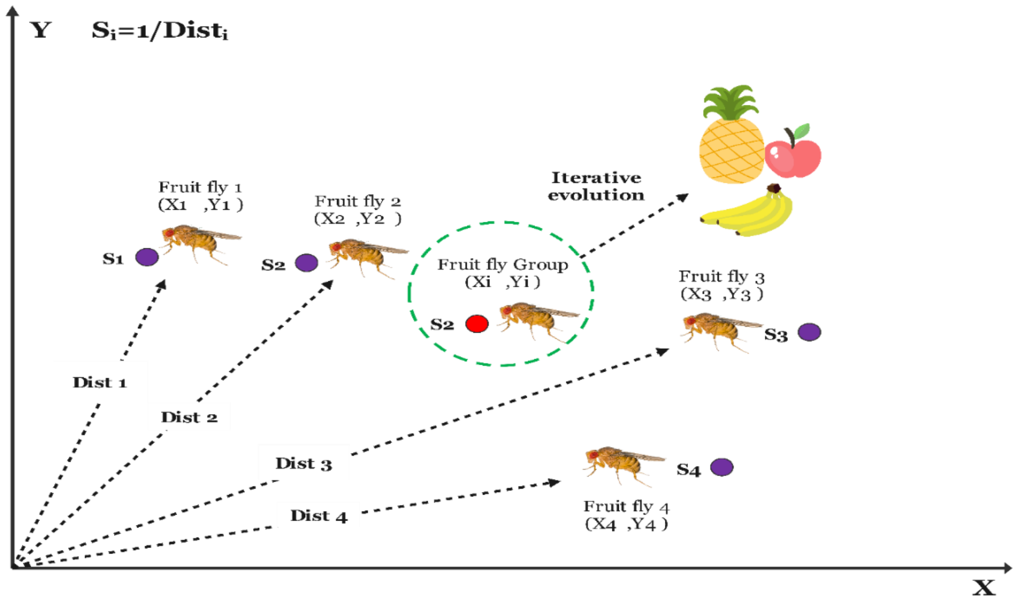

Figure 2: The trapping conduct shown by fruit flies

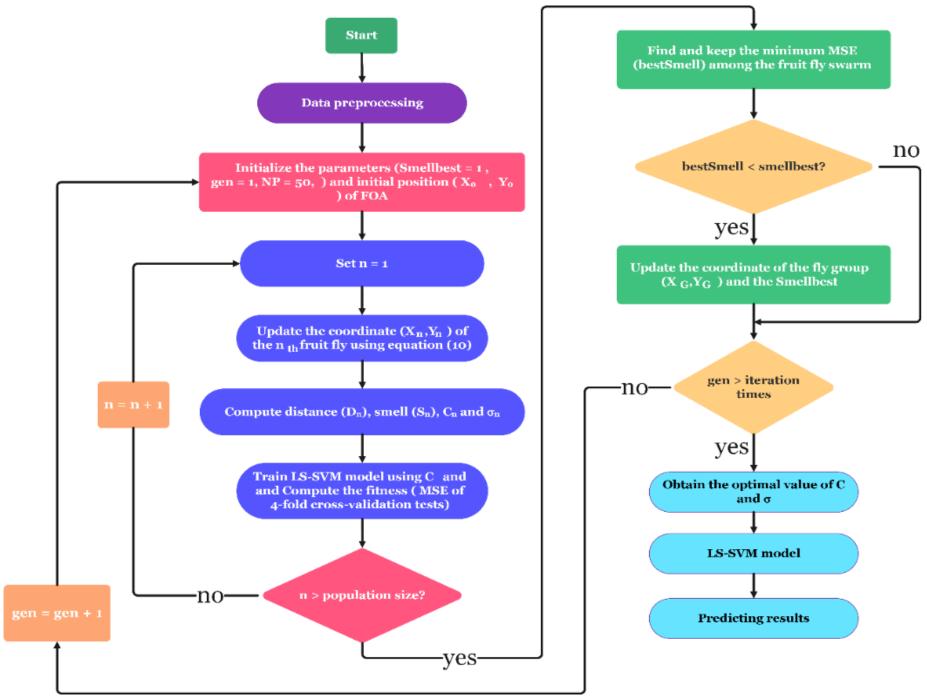

Figure 3: Flowchart of the Fruit Fly Optimization Algorithm to optimize the proposed model hyperparameters

4.3.2 Sparrow search algorithm

The SSA is a swarm enhancement technique that derives from the behavioral patterns of sparrows, shown in Fig. 4 [29]. The program categorizes the sparrow population into two distinct groups: producers and scroungers. Obtaining food and controlling population mobility fall within the purview of the producers. Producers can get a greater amount of food when the fitness value improves. The scroungers, however, actively seek out the producers and endeavor to enhance the solutions offered by them. Within each cycle of the method, the producer's position updates are computed using a particular formula, which is not specified in the provided paragraph. In essence, the SSA algorithm attempts to discover the most advantageous solution for an optimization issue by emulating the collective intelligence, foraging, and defense mechanisms shown by sparrows. The updated locations of the producers are as follows:

$$X_{ij}^{i+1} = \begin{cases} X_{ij}^{t} \cdot \exp\left(\dfrac{-i}{\alpha \cdot \text{ iter}}\right) \text{ if } R_2 < ST \\ X_{ij}^{t} + Q \cdot L \text{ if } R_2 > ST \end{cases} \quad (10)$$

The provided equation has several variables and constants. The variable i ranges from 1 to N, with N being the size of the swarm. The variable t depicts the existing cycle. The variable j ranges from 1 to d, where d specifies the count of dimensions. The notation X_{1j}^{t} displays the value of the j-th dimension of the i-th sparrow at cycle t. The variable $\text{iter}_{\max}$ specifies the top range for the count of cycles. The constant α is a stochastic variable that takes on values between 0 and 1 with equal probability. The variable Q is a stochastic quantity that conforms to a Gaussian distribution. The matrix L is a $1 \times d$ matrix where all members are identical and equal to 1. The variables R_2 and ST indicate the alert value and the security cutoff, accordingly. The range of R_2 is from 0 to 1, whereas the range of ST is from 0.5 to 1.0. Regarding the scroungers, they observe the producers, and when they find a better source of food, they quickly abandon their existing location to compete intensely for the new resource. The equation dictating the adjustment of the scroungers' position is presented as follows:

$$X_{ij}^{i+1} = \begin{cases} Q \cdot \exp\left(\dfrac{X_{\text{monta}}^{t} - X_{ij}^{i}}{i^2}\right) & \text{if } i > n/2 \\ X_{p}^{i+1} + \left|X_{ij}^{t} - X_{p}^{t+1}\right| \cdot A^{+} \cdot L & \text{otherwise} \end{cases} \quad (11)$$

X_p was defined as the ideal position held by producers. The current worldwide worst location is considered to be the most unfavorable. The expression $A^{+} = AT(AA^{T})^{-1}$ describes the pseudo-inverse of matrix A, where A is a $1 \times d$ matrix with elements of 1 or -1 assigned at random.

When sparrows on the outside of the group feel danger, they quickly move to a safe region. Their conduct is driven by their desire to acquire a higher viewpoint. Conversely, sparrows positioned in the middle of the group exhibit a more unpredictable behavior, marked by aimless roaming, to get closer to surrounding sparrows. The mathematical model that governs this behavior may be formulated as follows:

$$X_{ij}^{t+1} = \begin{cases} X_{\text{best}}^{t} + \beta \cdot \left|X_{ij}^{t} - X_{brst}^{t}\right| & \text{if } f_i > f_g \\ X_{ij}^{t} + K \cdot \left(\dfrac{\left|X_{ij}^{t} - X_{\text{worrt}}^{t}\right|}{(f_i - f_w) + \varepsilon}\right. & \text{if } f_i = f_g \end{cases} \quad (12)$$

The above equation illustrates a mathematical formula for the Sum of Squared Differences (SSA) used in optimization situations. In this equation, X_{ber}^{t} depicts the current global optimum position, whereas X_{wors} indicates the existing global worst situation. The parameter β is a control variable that establishes the step size. It exhibits a normal dispersion with a mean of 0 and a variance of 1. K is a stochastic variable that falls inside the interval [-1,1]. The variables f_i and f_w represent the current sparrow's fitness values and the lowest fitness value, respectively. On the other hand, f_g displays the current global maximum fitness value. Furthermore, ε is the most minimal constant. The SSA utilizes this equation to choose the most favorable solution for an optimization issue.

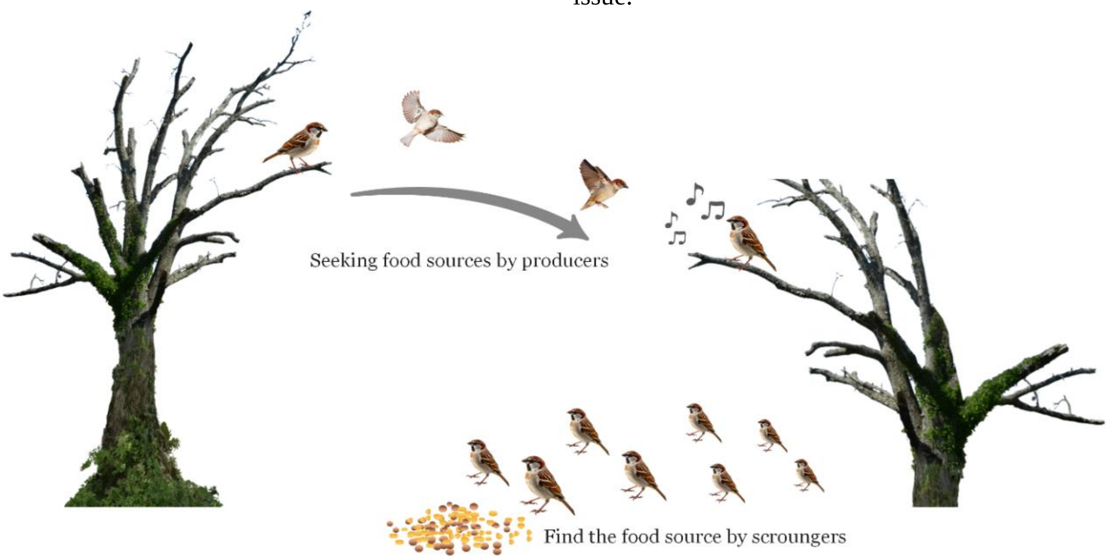

Figure 4: Behavioral diagram of the Sparrow Search Algorithm

4.3.3. Genetic algorithm

The genetic algorithm is a computational technique that mimics natural selection to solve optimization and search problems [30]. GA is composed of three fundamental components [31]. A set of textual or numeric symbols that each individual is assigned by the encoding entity is called a chromosome. The best encoding technique depends on the specific issue that needs to be resolved. The fitness function is also used to evaluate how accurately each person represented the answer. The fitness function was created specially to address the current problem. The evolutionary operators generate new individuals from preexisting ones through the processes of crossover, mutation, and selection. Crossover is a genetic process that combines the chromosomes of two distinct individuals to produce a new offspring. On the other hand, mutation causes random alterations to an individual's chromosomes. Selection is used to identify the individuals that reproduce the best.

4.3.4. Particle swarm optimization

Kennedy and Eberhart developed the PSO technique to address optimization issues. This method is based on the collective behavior observed in a swarm of particles [32]. Because the PSO technique tends to converge quickly and requires fewer parameters, it lowers computational overhead. Additionally, the likelihood of coming across a local solution that is not ideal is decreased by the thorough exploration carried out by several particles in pursuit of an ideal solution. Additionally, the algorithm features an efficient global search mechanism and is independent of derivatives. Each particle in the PSO searches a large search space for the optimal response. The search process begins with the random generation of candidate solutions, often called particles, in the search space. Particle velocities and fitness scores are often computed using a weighted mean of the number of features in the feature subset and the classification accuracy. This computation aids in updating the heading and velocity of their trajectories following the initial iteration, and the procedure is repeated until the stopping criterion is satisfied.

4.3.5. Northern goshawk optimization

NGO is a population-based metaheuristic inspired by the hunting mode of the northern goshawk (Accipiter gentilis), a very agile and efficient bird in hunting prey. The NGO algorithm replicates two principal stages of the goshawk's hunting process [33].

Major stages of the goshawk's hunting sequence:

(1) Prey identification and attack, and

(2) Pursuing and capturing prey through evasive flights.

Every agent (goshawk) of the population is a candidate solution to the optimization problem. The solutions are evaluated by an objective function and iteratively updated to converge to the optimum.

- Identifying and attacking prey.

As seen in Fig. 5, the northern goshawk haphazardly selects a victim during the first hunting phase and then quickly attacks it. This step improves the NGO's capability to explore by haphazardly selecting prey in the search domain. To choose the best site, this stage involves conducting a thorough search of the entire search domain. One way to articulate this is:

$$P_i = X_k, i = 1,2,\ldots,N, k = 1,2,\ldots\ldots,i-1,i+1,\ldots\ldots,N \tag{13}$$

$$x_{i,j}^{\text{new},P1} = \begin{cases} x_{i,j} + r\left(P_{i,j} - I x_{i,j}\right), F_{P_i} < F_i \\ x_{i,j} + r\left(x_{i,j} - P_{i,j}\right), F_{P_i} \geq F_i \end{cases} \tag{14}$$

$$X_i = \begin{cases} X_i^{\text{new},P1}, F_i^{\text{new},P1} < F_i \\ X_i, F_i^{\text{new},P1} \geq F_i \end{cases} \tag{15}$$

where: P_i displays the prey location of the i^{th} Northern goshawk, F_{P_i} is the value of its objective function, k is a haphazardly selected natural number from the range [1, N], $X_i^{\text{new},P1}$ depicts the updated status of the i^{th} recommended resolution, with its i^{th} dimension being $x_{i,j}^{\text{new},P1}$, $F_i^{\text{new},P1}$ indicates the value of the goal function based on the NGO's starting stage, and r and I are random numbers utilized to generate random behavior for the NGO during search and update. The variable (r) displays a haphazardly created number within $[0,1]$, whereas the variable (I) displays a haphazardly generated number within [1, 2].

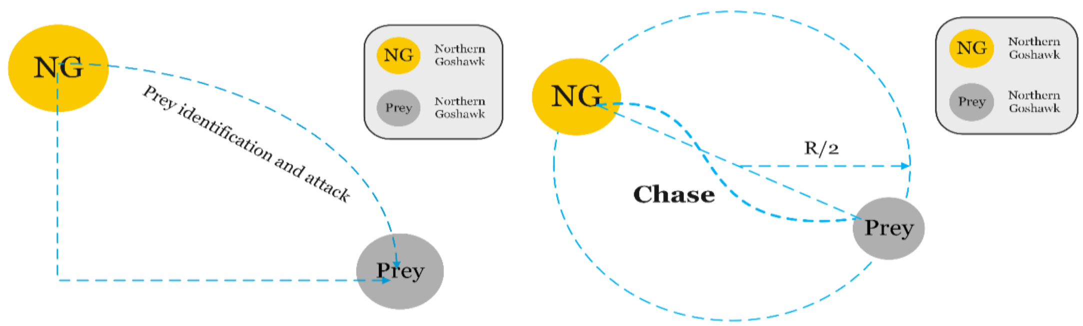

Figure 5: Prey identification and attack mechanism in Northern Goshawk Optimization

Phase (2): Engaging in pursuit and evasive maneuvers.

Following the attack, the goshawk pursues its escaping prey. This behavior is mimicked to make use of the search space locally to improve convergence. The mathematical representation of the ideas communicated in the second step may be expressed as Equations (16) - (18) in the following manner:

$$x_{i,j}^{\text{new},P2} = x_{i,j} + R(2r-1)x_{i,j} \tag{16}$$

$$R = 0.02\left(1 - \frac{t}{T}\right) \tag{17}$$

$$X_i = \begin{cases} X_i^{\text{new},P2}, F_i^{\text{new},P2} < F_i \\ X_i \quad\quad , F_i^{\text{new},P2} \geq F_i \end{cases} \tag{18}$$

The maximum cycle number is depicted as T, whereas the cycle counter is depicted as t. The parameter $X_i^{\text{new},\,P2}$ depicts the updated value of the i^{th} solution, and its dimension is $x_{i,j}^{\text{new}\ P2}$. The objective function derived from the second phase is depicted as $F_i^{\text{new,P2}}$. Fig. 6 illustrates the whole procedure of this optimizer.

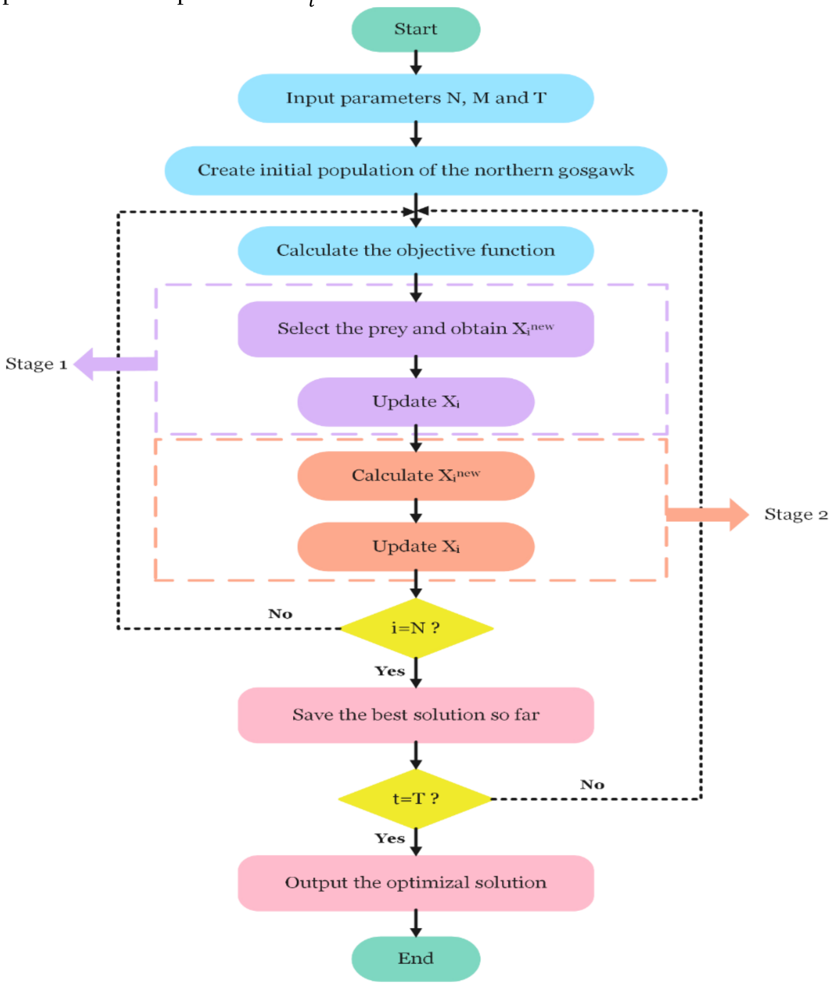

Figure 6: The NGO flowchart

The selection of the NGO was ultimately based on its biologically inspired two-phase optimization strategy, which successfully traverses the solution space by combining local exploitation (pursuit and attack) and global exploration (prey identification). This structure helps NGOs avoid overfitting and premature convergence, which are common issues in high-dimensional optimization. A scalable and suitable optimizer for nonlinear financial forecasting tasks that necessitate prompt decisions, like predicting the price of copper, an NGO can also benefit from its lightweight design, fewer control parameters, and fast convergence.

4.4 Model performance metrics

Metrics, or measurements, are utilized to review the productivity of a scheme, system, or process. The accuracy of the future estimates was evaluated based on many performance factors. Following a comprehensive investigation, these criteria were created to offer a thorough assessment of the estimates' dependability and correctness. Holdout data is required for trained machine learning schemes to perform properly to assess the difference between predicted and observed labels using various metrics. The metrics utilized to appraise the recommended scheme's prediction execution include R^2, relative square error (RSE), MAE, and MAPE.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{19}$$

$$\mathrm{MAPE} = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) \times 100 \tag{20}$$

$$\mathrm{RSE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(\bar{y} - \hat{y}_i)} \tag{21}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{22}$$

5 Outcomes

The primary goal of this study is to create and assess the best hybrid algorithm currently in use for precise copper price prediction. The objective of this endeavor was to provide dependable data to analysts and investors, empowering them to make knowledgeable financial assessments. This paper combined AdaBoost and optimizers like FOA, SSA, and NGO to review the volatility traits of international copper futures prices and the flexible link of influencing factor variables. The goal of this combo is to identify an algorithm that predicts copper prices more accurately and efficiently. This article uses daily international copper futures price data from January 2014 to October 2023. The data set used in this article was gathered utilizing an OHLC price and volume. Included in the assessment criteria are RSE, MAE, MAPE, and R^2. An Intel Core i7-10700KF processor operating at 3.80 GHz, paired with an NVIDIA GeForce GTX 3080ti graphics processing unit and 32 GB of RAM access memory, was used to implement the model in an advanced computing environment. Complex financial forecasting tasks can be handled efficiently with this setup. To improve computational efficiency and model performance, Python version 3.12.3 was used for the implementation, along with sophisticated libraries like NumPy (1.26.4), Pandas (2.2.3), Scikit-learn (1.5.2), and TensorFlow (2.17.0). Table 4 showcases the numerical assessment of the scheme's predictive precision.

Table 4: The outcomes of the proposed models' evaluation criteria.

Schemes/Metrics		AdaBoost	GA-Adaboost	PSO-Adaboost	FOA-AdaBoost	SSA-AdaBoost	NGO-AdaBoost
Train	R^2	0.9857	0.9864	0.9873	0.9893	0.9925	0.9978
	MAPE	1.6457	1.6261	1.5868	1.6860	1.2753	0.7138
	MAE	0.0519	0.0504	0.0498	0.0480	0.0371	0.0209
	RSE	0.1014	0.0984	0.0952	0.0872	0.0730	0.0400
Test	R^2	0.9836	0.9843	0.9856	0.9865	0.9906	0.9919
	MAPE	0.9655	0.9541	0.9468	0.9426	0.8124	0.6733
	MAE	0.0381	0.0376	0.0370	0.0368	0.0298	0.0268
	RSE	0.0702	0.0687	0.0659	0.0633	0.0542	0.0494

6 Discussion

Positive outcomes are obtained by using the recommended NGO-AdaBoost model in this work. Metaheuristic optimization is crucial for improving the performance of ensemble learning models, especially for financial time series forecasting, where data is often nonlinear, volatile, and prone to sudden regime changes. Five hybrid models that integrated AdaBoost with various metaheuristic optimization algorithms were assessed in this study: Particle Swarm Optimization (PSO), FOA, SSA, Genetic Algorithm (GA), and the proposed NGO. A detailed comparison of these hybrid models across training and testing datasets using four standard performance metrics R^2, MAPE, MAE, and RSE is given in Table 4 and fig. 7. The results show that, although with differing degrees of success, all optimizers enhanced AdaBoost's performance by modifying its hyperparameters. Though their optimization dynamics were less stable in high-dimensional search spaces, GA-AdaBoost, PSO-AdaBoost, and FOA-AdaBoost all achieved moderate gains, particularly in lowering MAE and RSE values. SSA-AdaBoost demonstrated superior exploration and convergence capabilities compared to the previously mentioned models, resulting in significantly better generalization on the test set. On the other hand, NGO-AdaBoost was the most dependable and effective. It achieved the best R^2 scores (0.9978 on training and 0.9919 on testing) and the lowest error values across all metrics. By combining local exploitation (pursuit and attack) with global exploration (prey identification), the NGO's biologically inspired two-phase optimization process successfully traverses the solution space, as evidenced by these results. This dynamic allows for precise adjustment of AdaBoost's parameters, resulting in dependable forecasting even in highly variable and noisy market conditions. In conclusion, the NGO-AdaBoost hybrid model exhibits remarkable predictive accuracy, stability, and efficiency in addition to outperforming conventional optimization-enhanced AdaBoost variants. Because of these characteristics, it is an extremely useful and appropriate tool for strategic decision-making in industries that rely heavily on commodities. Having a distinct AdaBoost model incorporated allows for the separation and measurement of the unique contribution of each optimizer to enhancing model accuracy. AdaBoost, as strong alone, was found to have limitations when hyperparameter tuning was performed, and its relatively lower prediction accuracy on all the metrics—R^2, MAE, MAPE, and RSE—was the outcome. The hybrid models did their work with observable gains, and NGO-AdaBoost achieved the best result, which indicated that blending NGO facilitated a more efficient search of the hyperparameter space. The performance table demonstrates that NGO-AdaBoost outperforms all other hybrid variants in terms of accuracy in both training and testing. FOA's search mechanism, based primarily on sensory flight, lacks adaptive ways to avoid being trapped in local minima and so is not suited for complicated parameter tuning tasks. SSA, with higher competitiveness compared to FOA, possessed stronger exploration but weaker exploitation. SSA exhibited a lower convergence rate and a more erratic search path, which led to poor tuning of ensemble components in noisy or highly nonlinear environments. Such inconsistency undermined its capacity for exploration-exploitation trade-off to achieve a balance between generalization and fit. Compared to it, the NGO, however, maintained a finer exploration-exploitation trade-off through its utilization of its two-stage (attack and pursuit) approach to both explore potential spots in the solution space and also optimize them in great detail. With this capacity, the NGO-AdaBoost hybrid model became a better generalizer and could perform more consistently as a price-predicting machine for copper prices. In conclusion, the NGO-AdaBoost hybrid model possesses excellent prediction accuracy, stability, and efficiency, as well as outperforms conventional optimization-augmented AdaBoost models. Because of these characteristics, it is an extremely practical and appropriate tool for strategic decision-making in highly commodity-reliant sectors. Additionally, an NGO's simple structure and fewer control parameters, which accelerate convergence and reduce computational complexity, make it more effective for real-time forecasting applications.

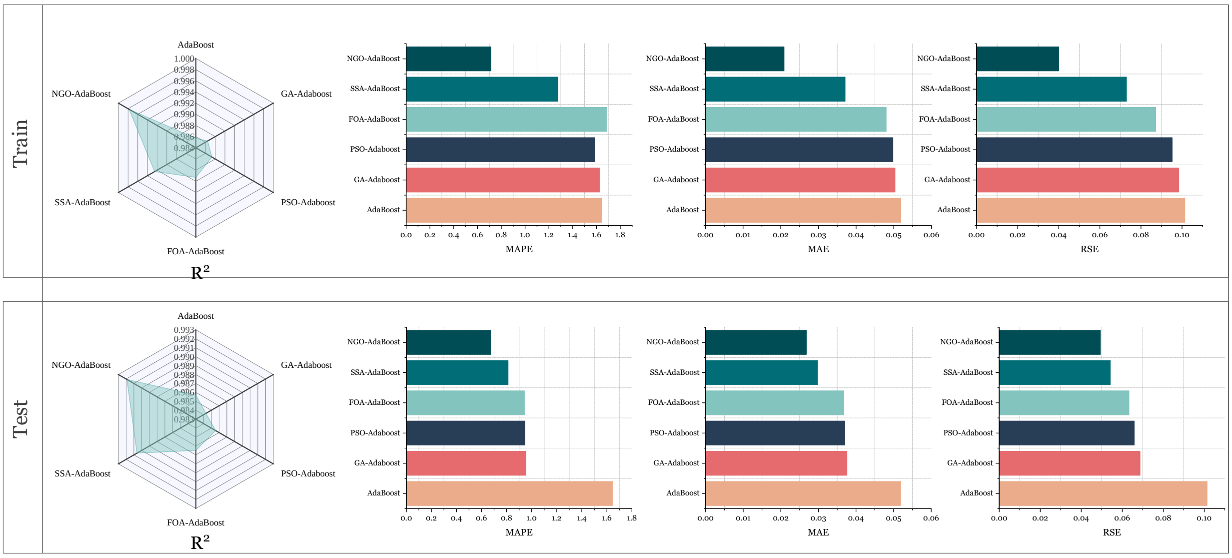

Figure 7: The model evaluation criteria's outcomes.

Figs. 8 and 9 show the actual and predicted value plots of NGO-AdaBoost on the training set and test set, respectively. The plots highlight the capacity of the model to accurately trace trends and ensure stability for learning as well as generalization purposes. The tight correspondence between predicted and actual curves, particularly on unseen test data, highlights the power of the model and its predictive accuracy.

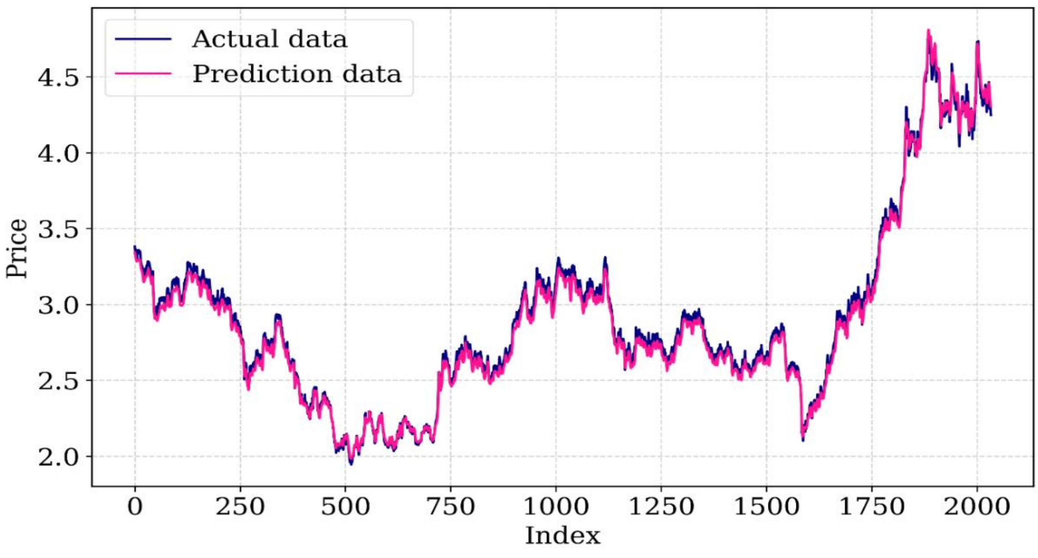

Figure 1: The train data prediction curve for NGO-AdaBoost

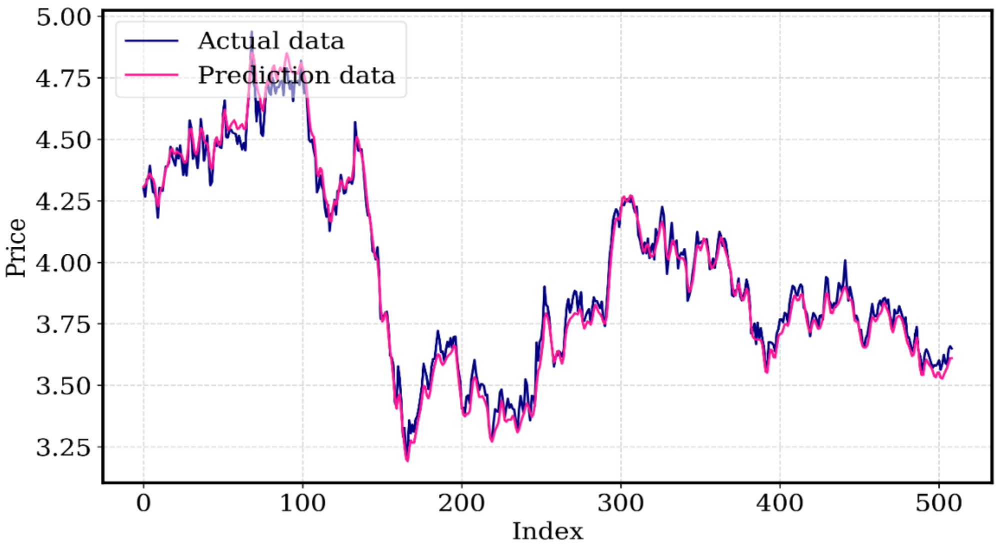

Figure 2: The test data prediction curve for NGO-AdaBoost

This research suggests many schemes such as AdaBoost, FOA-AdaBoost, SSA-AdaBoost, and FOA-AdaBoost to predict future copper prices. The findings reveal that NGO-AdaBoost surpasses other schemes regarding MAPE, RSE, MAE, and R^2 scores. Optimization of the NGO algorithm in connection with AdaBoost seems to be a better approach for copper price prediction. Because of the NGO-AdaBoost model's stability and computational efficiency, the model is well-suited for forecasting, which improves decision-making in sectors of the economy that depend heavily on copper; its precise prediction of copper prices gives investors and policymakers useful information that enables them to make informed decisions in a volatile market; and for copper producers, the model offers a useful tool for scheduling production and setting prices.

The individual models (ARIMA, SVR, GRU, Bi-LSTM, and XGBoost) have a relatively lower complexity as seen in Table 5. For example, models with relatively lower computational costs, like AdaBoost and ARIMA, are faster and more effective during training, but they usually perform worse for complex tasks where volatility and non-linearity are important considerations, like forecasting the price of copper. The AdaBoost model was chosen because of its robust ensemble learning feature, which can greatly enhance prediction quality and resilience when combined with optimization algorithms such as NGO. The hybrid models (GA-AdaBoost, PSO-AdaBoost, FOA-AdaBoost, SSA-AdaBoost, and NGO-AdaBoost) all show increased computational complexity as a result of the optimization step integrated with AdaBoost. Metaheuristic optimizers used to adjust the hyperparameters of the AdaBoost model also increase and improve training time significantly. Particularly, with greater training complexity than less complex models, NGO-AdaBoost achieves improved performance in recognizing complex copper price prediction patterns. Improved accuracy, reduced error rates, and overall NGO-AdaBoost model performance in copper price prediction compensated for the increased complexity of hybrid models. In short, the NGO-AdaBoost model provides a trade-off between precision and computational expense. Though more computationally expensive than conventional models, it is a very potent model for forecasting copper prices because of its superior predictive power, reduced error margins, and enhanced performance on unforeseen data. Its computational expense is justified in terms of the more accurate and stable performance of the model in an extremely volatile market.

Table 5. Training complexity of various models.

Training complexity (FLOPs*)	ARIMA	SVR	GRU	Bi-LSTM	XGBoost	AdaBoost	GA-Adaboost	PSO-Adaboost	FOA-AdaBoost	SSA-AdaBoost	NGO-AdaBoost
	4.03E+11	4.15E+11	4.86E+11	7.67E+11	5.55E+11	4.61E+11	1.87E+12	1.42E+12	2.68E+12	2.63E+12	1.49E+12

6.1 Comparative evaluation of the proposed model with state-of-the-art techniques

To confirm the performance of the proposed AdaBoost model, this paper compared it with five current state-of-the-art prediction methods: Autoregressive Integrated Moving Average (ARIMA), Support Vector Regression (SVR), Gated Recurrent Unit (GRU), Bidirectional Long Short-Term Memory (Bi-LSTM), and Extreme Gradient Boosting (XGBoost). Table 6 and fig 10 show the summarization results using four traditional evaluation metrics: R², MAPE, MAE, and RSE, for training and test sets.

Table 6: Comparing the outcomes of the proposed Adaboost model's evaluation criteria with the state-of-the-art methods.

Schemes/Metrics		ARIMA	SVR	GRU	Bi-LSTM	XGBoost	AdaBoost
Train	R^2	0.9399	0.9534	0.9703	0.9766	0.9805	0.9857
	MAPE	4.4475	3.6715	2.7182	2.4089	1.9231	1.6457
	MAE	0.1314	0.0919	0.0778	0.0714	0.0620	0.0519
	RSE	0.2080	0.1825	0.1455	0.1293	0.1185	0.1014
Test	R^2	0.9372	0.9502	0.9675	0.9751	0.9785	0.9836
	MAPE	2.2071	2.0387	1.3553	1.1973	1.2118	0.9655
	MAE	0.0888	0.0767	0.0521	0.0499	0.0466	0.0381
	RSE	0.1375	0.1217	0.0988	0.0866	0.0804	0.0702

It produced the greatest R² values (training 0.9857, testing 0.9836), which indicates a superior ability to account for the variance of copper price trends. It produced the lowest MAPE and MAE, high precision, and the lowest variation. The lowest RSE values indicate a more central distribution of errors and a superior fit to actual values. These results validate the fact that AdaBoost is a highly effective and stable model for predicting copper prices, better than traditional statistical models (ARIMA, SVR) and deep learning models (GRU, Bi-LSTM). Based on this strong baseline performance, AdaBoost was selected as the baseline ensemble learner to be further optimized through metaheuristic optimization. Its combination with an NGO at the level of optimization enabled hyperparameter tuning (learning rate, number of estimators), further enhancing generalization performance in especially noisy or nonlinear market conditions.

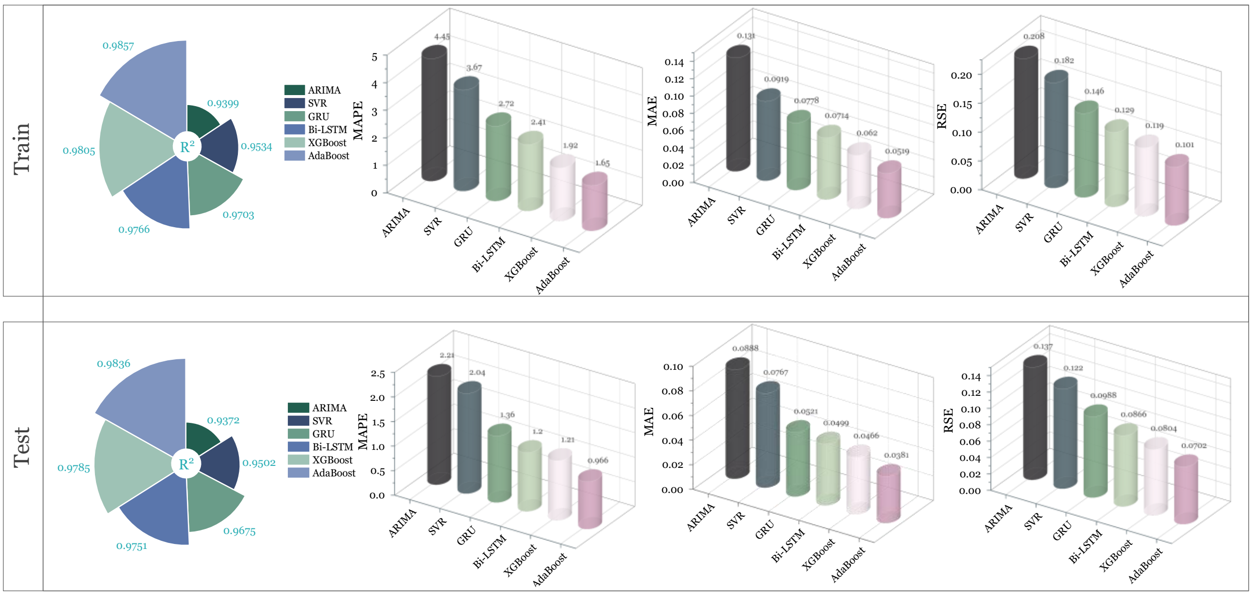

Figure 10: Contrasting the results of the suggested Adaboost model with state-of-the-art techniques.

6.2 Statistical Significance tests

The models' statistical significance was assessed by comparing the NGO-AdaBoost model's performance to that of other benchmark models using the Wilcoxon signed-rank test. Table 7 displays the Wilcoxon test results, test statistics (Wilcoxon values), and corresponding p-values for each model. The p-values indicate the statistical significance, whereas the Wilcoxon values show the performance differences between the paired models. A p-value of less than 0.05 indicates a statistically significant difference in performance. The table shows that the NGO-AdaBoost model performs better than AdaBoost, GA-AdaBoost, PSO-AdaBoost, FOA-AdaBoost, and SSA-AdaBoost in terms of predictive accuracy. Additionally, its p-value (1.36E-67) is the lowest.

Table 7: Wilcoxon Test Statistics and P-values for Model Comparison.

Models	Wilcoxon	P-value
AdaBoost	18506	2.07E-17
GA-Adaboost	20233	1.95E-13
PSO-Adaboost	19518	1.39E-09
FOA-AdaBoost	31227	2.72E-07
SSA-AdaBoost	2493	1.17E-65
NGO-AdaBoost	1358	1.36E-67

6.3 Comparison with literature reviews

The literature reviews for copper price prediction are compared in Table 8 and Fig.11, where the accuracy of various models and approaches is compared with the model suggested in this study. These findings indicate that the suggested model is significantly has more accuracy for predictions. These findings offer insightful information that helps investors make well-informed investment decisions.

Table 8: Performance comparison between the suggested model in this study and the methods and models offered in literature reviews.

Methods	Results	Year	Reference
RBF	R^2: 0.727	2024	[34]
RF	R^2: 0.608	2024	
SVM	R^2: 0.805	2024	
Deep Extreme Learning Machine (DELM)	R^2 : 0.959	2024	
XGB	R^2: 0.935	2024	[35]
GEP	R^2: 0.947	2024	
SSO-XGB	R^2 : 0.962	2024	
HHO-XGB	R^2: 0.959	2024	
LSTM	R^2 : 0.99012	2023	[19]
ANFIS-ACO	R^2 : 0.956	2021	[36]
NGO-Adaboost	**R^2: 0.9919**	**Proposed model**	

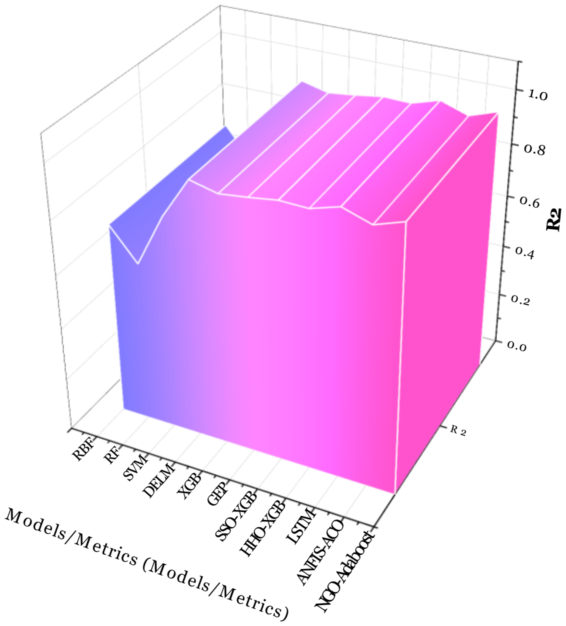

Figure 11: Comparison of the performance of the proposed model with other models presented in the literature reviews for predicting the price of copper.

7 Conclusions

Accurately predicting copper prices in both the spot and futures markets provides significant benefits to several stakeholders. Increased price predictability may allow policymakers to control the market more effectively. Copper producers can increase the efficiency and responsiveness of their production processes. As a result, investors may more effectively develop profitable investment plans for the short, medium, or long term. The purpose of this research is to forecast copper prices using a hybrid forecasting strategy that combines optimal schemes and ensemble learning algorithms. This work aims to fill the lack of research on copper price anticipation by providing a hybrid technique that overcomes the constraints of conventional time series schemes.

The main findings obtained from this investigation are as follows:

To forecast copper prices, this study introduced a new hybrid forecasting model that combines NGO and AdaBoost. This research uses historical data, including OHLC price and volume. The January 2014–October 2023 dataset period is used. The findings demonstrated that the NGO-AdaBoost model outperformed other metaheuristic-based models like PSO and GA-AdaBoost as well as more conventional machine learning models like SVR and ARIMA. Out of all the models tested, the NGO-AdaBoost model produced the best predictions with the highest R2 values (0.9978 training and 0.9919 testing) and the lowest error levels (MAPE: 0.6733, MAE: 0.0268). This resulted from the NGO's ability to successfully modify AdaBoost's hyperparameters to trade off local exploitation and global search through its biologically inspired two-phase search process.

. The outcomes display the robustness of the model in even adverse, noisy economic climates like predicting copper prices. The data clearly shows that the NGO-AdaBoost model works better than any other model. It also produces precise forecasts. Forecasts might be helpful to investors, policymakers, and manufacturers.

Notwithstanding its high predictive accuracy, the NGO-AdaBoost model has a few limitations. The model's sensitivity to AdaBoost and NGO hyperparameter settings may necessitate considerable fine-tuning to attain peak performance. NGOs help automate this process, but improper configurations can still result in less-than-ideal results. As with most ensemble methods, there is also a risk of overfitting, particularly when applied to small or noisy datasets. The model's performance has only been validated using data on copper prices; its applicability to other commodities or financial assets has not yet been established. Ultimately, using metaheuristic optimization increases computational complexity, which may limit scalability for real-time or large-scale applications.

Subsequent studies will build on this research by assessing the performance of various meta-heuristic optimization algorithms, such as Simulated Annealing (SA), Artificial Bee Colony (ABC), and Ant Colony Optimization (ACO), in enhancing AdaBoost for copper price forecasting. The application of external factors like macroeconomic factors (such as inflation rates, interest rates, and commodity prices) and geopolitical events (such as trade wars and policy changes) will also be critical in enhancing the robustness of the model. Taking these factors into account is most likely going to increase the model's ability to predict due to their crucial role in variations in copper price. Additionally, learning more advanced ensemble methods like Gradient Boosting Machines (GBM) will give a well-rounded understanding of the best techniques for predicting prices. Finally, using techniques like SHAP or LIME, the explainability of NGO-AdaBoost can be improved such that stakeholders can understand the driving factors behind the predictions of the model and have faith in its outcomes in real-world applications.

Acknowledgment

This investigation was backed by a teaching research project of Hubei Polytechnic University: "Exploration and Practice on the Cultivation of the 'Three Creations' Ability of Local College Talents Based on the Industrial College under the Background of New Engineering" (No.2024004).

Authorship contribution statement

Xianming Liu: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

References

[1] Dehghani, H (2018). Forecasting copper price using gene expression programming. *Journal of Mining and Environment*, Shahrood University of Technology, 9(2), pp. 349–360. https://doi.org/10.22044/jme.2017.6195.1435.

[2] Dehghani, H. and D. Bogdanovic (2018). Copper price estimation using bat algorithm. *Resources Policy*, Elsevier, 55, pp. 55–61. https://doi.org/10.1016/j.resourpol.2017.10.015.

[3] Lasheras, F.S., F.J. de Cos Juez, A.S. Sánchez, A. Krzemień and P.R. Fernández (2015). Forecasting the COMEX copper spot price by means of neural networks and ARIMA models. *Resources Policy*, Elsevier, 45, pp. 37–43. https://doi.org/10.1016/j.resourpol.2015.03.004.

[4] Liu, C., Z. Hu, Y. Li and S. Liu (2017). Forecasting copper prices by decision tree learning. *Resources Policy*, Elsevier, 52, pp. 427–434. https://doi.org/10.1016/j.resourpol.2017.05.007.

[5] Hu, C., X. Liu, B. Pan, H. Sheng, M. Zhong, X. Zhu and F. Wen (2017). The impact of international price shocks on China's nonferrous metal companies: A case study of copper. *Journal of Cleaner Production*, Elsevier, 168, pp. 254–262. https://doi.org/10.1016/j.jclepro.2017.09.035.

[6] Cheng, C., A. Sa-Ngasoongsong, O. Beyca, T. Le, H. Yang, Z. Kong and S.T.S. Bukkapatnam (2015). Time series forecasting for nonlinear and non-stationary processes: a review and comparative study. *Iie Transactions*, Taylor & Francis, 47(10), pp. 1053–1071. https://doi.org/10.1080/0740817X.2014.999180.

[7] Kriechbaumer, T., A. Angus, D. Parsons and M.R. Casado (2014). An improved wavelet–ARIMA approach for forecasting metal prices. *Resources Policy*, Elsevier, 39, pp. 32–41. https://doi.org/10.1016/j.resourpol.2013.10.005.

[8] García, D. and W. Kristjanpoller (2019). An adaptive forecasting approach for copper price volatility through hybrid and non-hybrid models. *Applied Soft Computing*, Elsevier, 74, pp. 466–478. https://doi.org/10.1016/j.asoc.2018.10.007.

[9] Li, G. and Y. Li (2015). Forecasting copper futures volatility under model uncertainty. *Resources Policy*, Elsevier, 46, pp. 167–176. https://doi.org/10.1016/j.resourpol.2015.09.009.

[10] Chehal, D., P. Gupta and P. Gulati (2023). Predicting the Usefulness of E-Commerce Products' Reviews using Machine Learning Techniques. *Informatica*, Slovenian Society Informatika, 47(2). https://doi.org/10.31449/inf.v47i2.4155.

[11] Zhang, T. and X. Yang (2024). Energy-Saving Design of Smart City Buildings Based on Deep Learning Algorithms and Remote Sensing Image Scenes. *Informatica*, Slovenian Society Informatika, 48(19). https://doi.org/10.31449/inf.v48i19.6049.

[12] Jelenčič, J. and D. Mladenić (2022). Improving modeling of stochastic processes by smart denoising. *Informatica*, Slovenian Society Informatika, 46(1). https://doi.org/10.31449/inf.v46i1.3875.

[13] Barstuğan, M. and R. Ceylan (2020). The effect of dictionary learning on weight update of AdaBoost and ECG classification. *Journal of King Saud University-Computer and Information Sciences*, Elsevier, 32(10), pp. 1149–1157. https://doi.org/10.1016/j.jksuci.2018.11.007.

[14] Ferguson-Lees, J (2001). *Raptors of the World*. Houghton Mifflin Company.

[15] El-Dabah, M.A., R.A. El-Sehiemy, H.M. Hasanien and B. Saad (2023). Photovoltaic model parameters identification using Northern Goshawk Optimization algorithm. *Energy*, Elsevier, 262, pp. 125522. https://doi.org/10.1016/j.energy.2022.125522.

[16] García-Gonzalo, E., P.J. García Nieto, J. Gracia Rodríguez, F. Sánchez Lasheras and G. Fidalgo Valverde (2023). A support vector regression model for time series forecasting of the COMEX copper spot price. *Logic Journal of the IGPL*, Oxford Academic, 31(4), pp. 775–784. https://doi.org/10.1093/jigpal/jzac039.

[17] Chen, J., J. Yi, K. Liu, J. Cheng, Y. Feng and C. Fang (2023). Copper price prediction using LSTM recurrent neural network integrated simulated annealing algorithm. *Plos One*, Public Library of Science (PLOS), 18(10), e0285631. https://doi.org/10.1371/journal.pone.0285631.

[18] Astudillo, G., R. Carrasco, C. Fernández-Campusano and M. Chacón (2020). Copper price prediction using support vector regression technique. *Applied Sciences*, MDPI, 10(19), p. 6648. https://doi.org/10.3390/app10196648.

[19] Zhao, Y., J. Chen, H. Shimada and T. Sasaoka (2023). Non-Ferrous Metal Price Point and Interval Prediction Based on Variational Mode Decomposition and Optimized LSTM Network. *Mathematics*, MDPI, 11(12), p. 2738. https://doi.org/10.3390/math11122738.

[20] Luo, H., D. Wang, J. Cheng and Q. Wu (2022). Multi-step-ahead copper price forecasting using a two-phase architecture based on an improved LSTM with novel input strategy and error correction. *Resources Policy*, Elsevier, 79, p. 102962. https://doi.org/10.1016/j.resourpol.2022.102962.

[21] Khoshalan, H.A., J. Shakeri, I. Najmoddini and M. Asadizadeh (2021). Forecasting copper price by application of robust artificial intelligence techniques. *Resources Policy*, Elsevier, 73, p. 102239. https://doi.org/10.1016/j.resourpol.2021.102239.

[22] Wang, L., Y. Guo, M. Fan and X. Li (2022). Wind speed prediction using measurements from neighboring locations and combining the extreme learning machine and the AdaBoost algorithm. *Energy Reports*, Elsevier, 8, pp. 1508–1518. https://doi.org/10.1016/j.egyr.2021.12.062.

[23] Obiedat, R. and S.A. Toubasi (2022). A combined approach for predicting employees’ productivity based on ensemble machine learning methods. *Informatica*, Slovenian Society Informatika, 46(5). https://doi.org/10.31449/inf.v46i5.3839.

[24] Lu, J., H. Hu and Y. Bai (2015). Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and AdaBoost algorithm. *Neurocomputing*, Elsevier, 152, pp. 305–315. https://doi.org/10.1016/j.neucom.2014.10.065.

[25] Peng, T., J. Zhou, C. Zhang and Y. Zheng (2017). Multi-step ahead wind speed forecasting using a hybrid model based on two-stage decomposition technique and AdaBoost-extreme learning machine. *Energy Conversion and Management*, Elsevier, 153, pp. 589–602. https://doi.org/10.1016/j.enconman.2017.10.021.

[26] Wu, X. and S. Meng (2016). E-commerce customer churn prediction based on improved SMOTE and AdaBoost, In *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, IEEE, Kunming, pp: 1–5. https://doi.org/10.1109/ICSSSM.2016.7538581.

[27] Zhao, H., H. Yu, D. Li, T. Mao and H. Zhu (2019). Vehicle accident risk prediction based on AdaBoost-SO in VANETs. *IEEE Access*, IEEE, 7, pp. 14549–14557. https://doi.org/10.1109/ACCESS.2019.2894176.

[28] Sutradhar, S., S. Karforma, R. Bose and S. Roy (2023). A dynamic step-wise tiny encryption algorithm with fruit fly optimization for quality of service improvement in healthcare. *Healthcare Analytics*, Elsevier, 3, p. 100177. https://doi.org/10.1016/j.health.2023.100177.

[29] Xue, J. and B. Shen (2020). A novel swarm intelligence optimization approach: sparrow search algorithm. *Systems Science & Control Engineering*, Taylor & Francis, 8(1), 22–34. https://doi.org/10.1080/21642583.2019.1708830.

[30] Mitchell, M (1998). *An introduction to genetic algorithms*. MIT press.

[31] Alkafaween, E., A.B.A. Hassanat and S. Tarawneh (2021). Improving initial population for genetic algorithm using the multi linear regression based technique (MLRBT). *Communications-Scientific Letters of the University of Zilina*, Žilinská univerzita v Žilině, 23(1), E1–E10.

https://www.ceeol.com/search/article-detail?id=1120555.

[32] Kennedy, J. and R. Eberhart (1995). Particle swarm optimization, In *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, Perth, WA, Australia, pp: 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

[33] Sadeeq, H.T. and A.M. Abdulazeez (2022). Improved northern Goshawk optimization algorithm for global optimization, In *2022 4th International Conference on Advanced Science and Engineering (ICOASE)*, IEEE, Zakho, Iraq, pp: 89–94. https://doi.org/10.1109/ICOASE56293.2022.10075576.

[34] Li, N., J. Li, Q. Wang, D. Yan, L. Wang and M. Jia (2024). A novel copper price forecasting ensemble method using adversarial interpretive structural model and sparrow search algorithm. *Resources Policy*, Elsevier, 91, pp. 104892. https://doi.org/10.1016/j.resourpol.2024.104892.

[35] Nabavi, Z., M. Mirzehi and H. Dehghani (2024). Reliable novel hybrid extreme gradient boosting for forecasting copper prices using meta-heuristic algorithms: A thirty-year analysis. *Resources Policy*, Elsevier, 90, p. 104784. https://doi.org/10.1016/j.resourpol.2024.104784.

[36] Khoshalan, H.A., J. Shakeri, I. Najmoddini and M. Asadizadeh (2021). Forecasting copper price by application of robust artificial intelligence techniques. *Resources Policy*, Elsevier, 73, pp. 102239. https://doi.org/10.1016/j.resourpol.2021.102239.