Research Article

Iyad Abu Doush, Basem Ahmed, Mohammed A. Awadallah, Mohammed Azmi Al-Betar*, and Noor Aldeen Alawad

# Improving multilayer perceptron neural network using two enhanced moth-flame optimizers to forecast iron ore prices

https://doi.org/10.1515/jisys-2023-0068
received May 24, 2023; accepted October 04, 2023

**Abstract:** The quality of the output produced by the multi-layer perceptron neural network depends on the careful selection of its weights and biases. The gradient descent technique is commonly used for choosing MLP's optimal configuration, but it can suffer from being stuck in local optima and slow convergence toward promising regions in the search space. In this article, we propose two new optimization algorithms based on the moth-flame optimization algorithm (MFO), which mimics moths' special navigation methods at night. We use these algorithms to enhance the performance of the training process of the MLP neural network. To demonstrate the effectiveness of our approach, we apply it to the problem of predicting iron ore prices, which plays an important role in the continuous development of the steel industry. We use a large number of features to predict the iron ore price, and we select a promising set of features using two feature reduction methods: Pearson's correlation and a newly proposed categorized correlation. Surprisingly, new features not mentioned in the literature are discovered, and some are discarded. The time series dataset used has been extracted from several sources and pre-processed to fit the proposed model. We compare our two proposed MFO algorithms, the roulette wheel moth-flame optimization algorithm and the global best moth-flame optimization algorithm, against four swarm intelligence algorithms and five classical machine learning techniques when predicting the iron ore price. The results acquired indicate the superior performance of the suggested algorithms concerning prediction accuracy, root-mean-square error, mean-square error, average absolute relative deviation, and mean absolute error. Overall, our work presents a promising approach for improving the performance of MLP neural networks, and it demonstrates its effectiveness in the challenging problem of predicting iron ore prices.

**Keywords:** multilayer perceptron neural network, iron ore price prediction, predicting, training neural network, swarm intelligence optimizers, moth-flame algorithm

*** Corresponding author: Mohammed Azmi Al-Betar,** Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 61001, United Arab Emirates; Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, P.O. Box 50, Al-Huson, Irbid 21110, Jordan, e-mail: m.albetar@ajman.ac.ae

**Iyad Abu Doush:** Department of Computing, College of Engineering and Applied Sciences, American University of Kuwait, Salmiya 20002, Kuwait; Computer Science Department, Yarmouk University, Irbid 21163, Jordan, e-mail: eyad_win@yahoo.com

**Basem Ahmed:** Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza, Palestine

**Mohammed A. Awadallah:** Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza, Palestine; Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 61001, United Arab Emirates

**Noor Aldeen Alawad:** Computer Science Department, Yarmouk University, Irbid 21163, Jordan

# 1 Introduction

In the domain of industrial production, iron ore plays a foundational role, serving as a vital raw material for the steel manufacturing sector [1]. This sector, in turn, drives the creation of buildings and infrastructure, which are fundamental components of modern development. The essential nature of iron ore in steel production gives rise to a significant mutual dependence between the two resources, where fluctuations in the price of iron ore have a ripple effect throughout the steel industry. The complexities of this relationship are encapsulated in the observation that changes in iron ore prices can lead to varying impacts on steel prices, with estimates suggesting that alterations in iron ore prices can influence steel prices by approximately 10–20%.

Iron ore's global pricing landscape is characterized by its susceptibility to substantial fluctuations, a dynamic nature that bears the potential to impact countries engaged in the import or export of this precious resource [2]. A conspicuous illustration of this phenomenon is the staggering 784% surge in China's demand for iron ore between 2000 and 2010 [3]. This surge is emblematic of the resource's centrality in underpinning the monumental growth of economies through infrastructure expansion.

The volatility in iron ore pricing, notably commencing around 2008, introduces a significant challenge to the industry's stability [4]. This price instability raises concerns, particularly within the steel sector, as it directly impacts the feasibility of securing the necessary raw materials through long-term contracts. Currently, iron ore pricing hinges on agreements between major consumers and producers, with prominent producers including Vale in Brazil, Rio Tinto in the United Kingdom, and B.H.P Billiton in Australia. Given that approximately 1.5 tons of iron ore yield 1 ton of unpurified steel [5].

Given these intricate interplays, the effort to precisely forecast the future price of iron ore becomes a primary concern for stakeholders within the steel industry. A skillful predictive model holds the potential to optimize resource procurement strategies, reduce unnecessary expenditure, and facilitate well-informed decisions regarding the establishment of long-term agreements with iron ore suppliers. The task, however, is not without challenges. Iron ore price prediction is inherently intricate, involving the synthesis of multifaceted inputs, the discernment of nuanced patterns, and the management of inherent uncertainties. Moreover, traditional methods, such as those based on gradient descent learning mechanisms, while valuable, contend with issues of slow convergence and vulnerability to local optima.

Developing an accurate model that can predict the future price of iron ore can help reduce the spending of the steel industry when obtaining raw materials. Such information can help the stakeholders to make critical decisions about signing long-term contracts with the iron ore suppliers [6]. Swarm intelligence optimizers depend on developing several solutions stored in the population. These solutions are mixed and modified until the stop condition is met. The population in the last evolution may converge prematurely. To overcome such a problem, the algorithms are enhanced by introducing operators that can help in reaching more promising regions in the search space [7].

Various nonlinear machine learning techniques are harnessed for solving resource prediction challenges, with the multi-layer perceptron (MLP) neural network being a prominent example. These techniques have been applied across diverse applications such as resource prediction [8–11]. The effectiveness of machine learning prediction models hinges on factors like input parameters, the structure of the model, and internal learning processes.

The optimization of weights and biases in the MLP often employs the gradient descent learning mechanism. However, this approach grapples with persistent issues: sluggish convergence and susceptibility to local optima. To address these challenges, researchers have proposed an enhancement to MLP prediction performance. This involves refining the prediction model's input parameters, encompassing weight and bias vectors, through the utilization of optimization techniques such as whale optimization algorithm [10], grasshopper optimization algorithm [1], etc.

The moth-flame optimization (MFO) algorithm is an innovative and nature-inspired optimization technique that draws inspiration from the intriguing relationship between moths and flames [12]. Just as moths are instinctively drawn toward light sources like flames, the MFO algorithm simulates this behavior to solve complex optimization problems. This algorithm belongs to the broader category of swarm intelligence (SI) algorithms, which mimic the collective behavior of natural swarms to find optimal solutions. Introduced as a novel optimization method, the MFO algorithm is characterized by its simplicity, efficiency, and ability to handle various types of optimization tasks. The algorithm imitates the natural behavior of moths in their navigation toward light sources. Moths tend to follow a specific pattern of movement while being attracted to a light source. They exhibit motion randomness as well as attraction toward the light. Therefore, MFO has been utilized to tackle various kinds of optimization problems as summarized in the study by Sahoo et al. [13].

This article presents the use of two new MFOs to enhance the performance of MLP to provide a more accurate prediction of the iron ore price, which are the roulette wheel moth-flame optimization algorithm (RWMFO) and global best moth-flame optimization algorithm (GBMFO). The effectiveness of the two proposed algorithms in predicting the iron ore price is evaluated by comparing them against four swarm intelligence algorithms. The comparative algorithms are Harris Hawks optimizer (HHO) [14], Jaya optimizer [15], particle swarm optimizer (PSO) [16], and the original MFO. The proposed algorithms are then compared against five machine-learning techniques. The results acquired indicate the superior performance of the suggested two algorithms concerning prediction accuracy and reducing the mean-square error.

This article provides several contributions, which can be summarized as follows:

- This study compares two feature selection techniques' impact on the performance of the proposed algorithms when predicting the price of iron ore: first, analyzing correlation through Pearson's correlation (called **Set of attributes within Group A**) and second, the correlation categorized within a proposed framework (called **Set of attributes within Group B**).
- To the best of our knowledge, this research is the first attempt to apply the following swarm intelligence techniques as MLP optimizers to predict the monthly iron ore prices HHO, JAYA, PSO, and MFO.
- Two new MFOs are proposed and applied as MLP optimizers to predict the iron ore prices that are RWMFO and GBMFO.
- Comparisons were conducted between the performance of the suggested models and five conventional machine learning models.

The rest of this article is organized as follows: the literature is presented in Section 2. The multilayer perceptron (MLP) neural network, the SI techniques applied, and the proposed model are demonstrated in Section 3. The used dataset and the obtained results are presented in Section 4. Finally, the conclusions and future research directions are outlined in Section 5.

# 2 Related work

Several techniques are proposed to forecast the price of a wide range of commodities. In this section, we will give a review of relevant methods used for price forecasting, which can be categorized as neural networks and hybrid models.

## 2.1 Predict commodity price using neural network

Various recent studies have been conducted using artificial neural networks (ANNs) to predict commodity prices [17–20]. Kristjanpoller et al. [21] analyzed the main Latin-American markets and found that ANN models were able to improve forecasting performance compared to generalized auto regressive conditional heteroskedasticity (GARCH) models with robust results.

Fath et al. [22] developed two popular ANN methods, the multilayer perception (MLP) and radial basis function (RBF), to predict the future costs of crude oil. The results show that RBF obtains higher accuracy compared to the MLP model. Ramyar and Kianfar [23] proposed a neural network model in an attempt to estimate the oil price predictability. The crude oil prices could be affected by different factors like the limited supply of crude oil and the continuous changes in currency policies. Through empirical analysis, it is demonstrated that oil prices can be predicted with a high degree of accuracy by combining various factors into the model: neural network design, feature engineering, and the market realities regarding crude oil.

Kim [24] proposed a genetic algorithm (GA)-based approach to select instances in ANNs to mine financial data. The GA optimizes the weights of connections between layers and the associated instance selection task. The suggested model is used for stock market analysis. According to experimental data, the GA technique is a potential strategy, for instance, selection in ANN. Bildirici and Ersin [25] developed a new variation of linear GARCH, fractionally integrated (FI-GARCH), and asymmetric power (APGARCH) models with logistic smooth transition autoregressive (LSTAR)-type nonlinearity modeling. ANNs have been added to the new models to get the advantage of the capabilities of learning and forecasting. The multilayer perceptron (MLP) ANN model and the LSTAR model have significant similarities in terms of their architecture. The results show that the LSTAR-based and neural network augmented models provide accurate forecasting of volatility in crude oil prices.

Livieris et al. [26] suggested a novel predictive deep learning model for forecasting the gold price. It has been proven that using long short-term memory (LSTM) layers with additional convolution layers can significantly improve its predictive performance. On the other hand, Primananda and Isa [27] analyzed and forecast gold prices by using a gated recurrent unit (GRU) model, which is another type of recurrent neural network (RNN). It is proven that GRU does better by presenting a sophisticated method for accurately forecasting gold prices.

Varma and Padma [28] proposed a model to predict agricultural commodity prices. It was found that models that use ANNs outperformed the models that use multiple linear regression for medium and long-term data. Specifically, for long-term predictions, the results demonstrate that neural network models can be highly effective.

Xu and Zhang [29] address various critical aspects of the neural network model, including algorithmic selection, delay configurations, hidden neuron counts, and data splitting ratios, to identify an optimal model configuration that delivers accurate and stable price forecasts. Through meticulous experimentation, they arrive at a model configuration featuring five delays and ten hidden neurons, powered by the Levenberg–Marquardt algorithm. The dataset is partitioned into training (80%), validation (10%), and testing (10%) subsets, yielding impressive relative root-mean-square errors (RMSEs) of 1.48, 1.49, and 1.47% for the respective phases. By revealing the efficiency of neural networks in the field of thermal coal price forecasting, the authors highlight a previously underexplored domain within the literature. The forecast outcomes generated by their model hold potential both as standalone technical predictions and as complementary inputs for policy analysis endeavors aimed at gaining insights into evolving price trends.

Xu and Zhang [30] incorporate a study that utilizes nonlinear autoregressive neural networks to forecast the wholesale price indices of canola and soybean oil in China over a 10-year period. The research assesses various model configurations, including algorithms, delays, hidden neurons, and data splitting ratios, to enhance accuracy and robustness. The findings demonstrate the effectiveness of this neural network approach, with models for canola and soybean oil prices achieving relative RMSEs ranging from 1.46 to 2.66%. These accurate and stable predictions hold practical implications for market participants and policymakers, serving as technical insights for decision-making and complementing fundamental forecasts for nuanced perspectives on price trends and policy formulation. Overall, the study underscores the value of neural networks in commodity price forecasting, bridging theory, and real-world applications.

Ding [31] explores forecasting commodity prices. This is crucial for macroeconomic decisions as well as micro-level management. The task involves unraveling the complex network of factors that exert influence. For this purpose, the article introduces a novel approach that combines the LSTM deep learning algorithm with external factors. This multifactor LSTM price prediction method leverages historical data using LSTM memory while incorporating the impact of external factors through a full connection layer. This innovative approach demonstrates improved accuracy and stability when compared to the BP neural network. In addition, the study involves analyzing commodity descriptions and price characteristics, identifying similar commodities, and using their historical price data to build a training set, thereby validating the proposed method's effectiveness in commodity price prediction.

In the context of comprehending commodity price projections, particularly within the metal business sector, the significance of accurate forecasts holds paramount importance for policymakers and a wide spectrum of market participants. According to this, a notable contribution to the field comes from the article titled "Enhancing Commodity price Forecasting in the Metal Industry: Evidence from Chinese Steel prices" by Xu and Zhang [32]. The article explores the field of forecast challenges inherent to the Chinese metal market, specifically concentrating on the steel sector. Spanning an extensive temporal span from 2010 to 2021, the research scrutinizes seven distinct regional steel prices encompassing various parts of China, including east, south, north, central south, northeast, southwest, and northwest regions. Employing nonlinear auto-regressive neural networks as the analytical tool, the study scrutinizes an array of factors, including training algorithms, hidden neurons, delay intervals, and data segmentation ratios, encompassing a total of 120 settings for each price series. Remarkably, the study unveils the development of a comparatively uncomplicated model that remarkably delivers not only high forecast accuracy but also robustness across all seven price series. Notably, the relative RMSE, a pivotal metric for forecast precision, remains consistently below 0.60% for the entirety of the examined series. The research findings propose a potent framework that can be adopted either in isolation or in conjunction with predictions from diverse models, serving as a valuable asset for evaluating price trends and conducting policy analysis for diverse categories of forecast users.

Joshi et al. [33] introduced an innovative method for enhancing the accuracy of metal price predictions. The approach centers around a neural network (NN) model that benefits from the guidance of an evolutionary algorithm. One of the standout features of this approach is its ability to automatically select optimal network parameters and architecture, contributing to a more adaptable and effective predictive model. To construct this model, a custom fitness function is devised, tailored to the unique attributes of time series metal price data. This function not only minimizes prediction errors but also strives to replicate the auto-correlation function inherent in the data. By harnessing average entropy values, the model optimally determines the count of input parameters, further refining its configuration. The article demonstrates the utility of this methodology through a case study involving gold price forecasting. Remarkably, the resulting evolutionary-based NN model yields specific parameter values, including a hidden node count of 9, a learning rate of 0.026, and a momentum value of 0.76, all contributing to accurate predictions. The method not only achieves the dual objectives of minimizing estimation errors and replicating auto-correlation but also exhibits superior performance compared to existing techniques in the field of metal price forecasting.

In summary, the reviewed studies collectively demonstrate the effectiveness of neural network models in predicting commodity prices across various markets. These models have the potential to provide accurate and stable forecasts, enriching decision-making processes for market participants, policymakers, and stakeholders. Despite challenges, ongoing innovations in neural network design and the incorporation of external factors continue to drive advancements in the field of commodity price forecasting.

## 2.2 Predict commodity price using hybridized neural network models

In the past few years, hybrid models have gained a lot of attention as a means of improving the models. This is because there is an imminent need to increase the accuracy of forecasts by achieving breakthroughs in the existing models. Furthermore, real-life problems entail many more factors than simply knowing the data. Khashei and Bijari [34] proposed hybrid approaches that are capable of solving these challenges by using nonlinear modeling as well as linear modeling.

ANN with LSTM played a significant role in stock market prediction [35–37]. The proposed models give better results than other traditional methods. Alameer et al. [38] presented a deep-learning model capable of predicting changes in coal prices. They proposed L LSTM and deep neural network (DNN) called LSTM–DNN. The results demonstrate that the hybrid LSTM–DNN model outperforms its competitors. Baffour et al. [39] integrated Glosten, Jagannathan, and Runkle (GJR) model into an ANN to forecast currency exchange rate volatility. According to the study, providing commodity price series can potentially improve model performance over different time horizons depending on the particular currency pair under study.

To forecast the monthly volatility in the copper market, García and Kristjanpoller [40] suggested a hybrid model called adaptive-GARCH-FIS that consists of a set of time series models. The experimental results show that the new model is powerful and makes predictions with the highest accuracy. In addition, Nguyen-Ky et al. [41] used a hybrid ANN-Bayesian for predicting seasonal water allocation prices with greater accuracy. The test results show that the proposed model outperforms the traditional ANNs.

Kristjanpoller and Minutolo [42] suggested a hybrid ANN-GARCH model with preprocessing to forecast the price volatility of bitcoin. The results show that the hybrid model tends to provide more accurate forecasts than individual forecasting models. In addition, Jannah et al. [43] proposed a hybrid ANN-GARCH model that combines both ANN and GARCH. This method was used to forecast the prices of rice, onion, red chili, and cayenne pepper in Jakarta. The results presented in the work show that the ANN-GARCH performs better than the autoregressive integrated moving average (ARIMA) in predicting the price of the four commodities. Moreover, with data on copper spot prices, Lasheras et al. [44] evaluated the performance of ARIMA and two ANNs (multilayer perceptron and Elman). In comparison with ARIMA, both neural network models were found to be superior.

Based on the prediction rule ensembles technique and DNNs, Manujakshi et al. [45] developed a hybrid stock prediction model. They found that the hybrid model of stock prediction is more accurate than individual prediction models, such as DNN and ANN, with an RMSE improvement of 5–7%. Ahmed et al. [46] proposed an innovative hybrid algorithm to forecast wind speed, which incorporates Krill Herd (KH) optimization algorithm and adaptive neuro-fuzzy inference system (ANFIS). The results show that the model can be used to predict wind speed effectively.

Kristjanpoller and Hernández [47] used a hybrid neural network model and GARCH models to anticipate the volatility of prices for gold, silver, and copper. The results demonstrate that using a hybrid neural network model to predict out-of-sample volatility significantly increases the predictive power of these three metals. Alameer et al. [10] developed a model for forecasting the monthly increases and decreases of gold prices over the long term. A whale optimization algorithm is used as a trainer to learn the multilayer perceptron NN. The work indicates that the hybrid model offers more accurate forecasts when compared with other models based on empirical evidence.

In the context of financial markets, achieving precise forecasts for stock price indices, given the pivotal role of volatility, remains crucial. To address this, a novel approach fusing multivariate artificial neural networks (MANNs) with dynamic conditional correlation (DCC)-generalized autoregressive conditional heteroscedastic (GARCH) models is proposed by Fatima and Uddin [48]. This hybrid model not only predicts stock price volatility but also dissects time-varying correlations. Empirical analysis, conducted on daily share price data from prominent stock markets including S&P 500 (USA), FTSE-100 (UK), KSE-100 (Pakistan), Malaysia (KLSE), and BSESN (India), covering January 1, 2013, to March 17, 2020, highlights the DCC-GARCH(1,1)-MANNs hybrid's superior forecast performance compared to the MANNs-DCC-GARCH(1,1) model. Notably, the DCC-GARCH(1,1)-MANNs hybrid excels not only in predictive accuracy but also in providing insights into correlations and volatility dynamics, distinguishing it from the MANNs-DCC-GARCH(1,1) hybrid. Consequently, this hybrid model emerges as the preferred choice for effective modeling and forecasting of stock price indices.

Addressing the volatility in rubber prices, Alzaeemi et al. [49] proposes a hybrid intelligent model to forecast rubber prices in Malaysia. Drawing on monthly pricing data from January 2016 to March 2021, the model combines algorithms including the radial basis functions neural network k-satisfiability logic mining (RBFNN-kSAT), along with factors like production, trade volumes, stock levels, exchange rates, and crude oil prices. Comparative analysis highlights the efficacy of the GWO with RBFNN-kSAT model, exhibiting an impressive 92% average accuracy and a strong correlation coefficient ($R$ = 0.983871), outperforming ABC with RBFNN-kSAT and PSO with RBFNN-kSAT models. Beyond its forecasting value, the research offers insights for policy decisions, aiding the Malaysian rubber industry in navigating price fluctuations and maintaining global market prominence.

Chen [50] advances the development of radial basis function neural networks (RBFnet) through computational intelligence and SI methods. By integrating artificial immune system (AIS) and ant colony optimization (ACO) approaches, the proposed hybrid AIS-ACO optimization (HIAO) algorithm harmonizes exploitation and exploration. This synergy enhances problem-solving by promoting solution space diversity and preventing premature convergence. Empirical results demonstrate the HIAO algorithm's efficacy in an accurate function approximation, spanning from theoretical nonlinear problems to practical applications like predicting crude oil spot prices.

Sharma et al. [51] address the intricacies of stock market forecasting by proposing an innovative hybrid approach that combines an ANN with a GA. These techniques are employed for the accurate prediction of two prominent US stock market indices, namely, DOW30 and NASDAQ100. Through meticulous data partitioning and validation, the study demonstrates the hybrid model's superiority over the singular backpropagation artificial neural network technique, showcasing enhanced predictive accuracy across both short- and long-term forecasting horizons. This research highlights the potential of integrated methodologies to effectively forecast in the field of nonlinear financial data.

Within the domain of time series forecasting, Hadwan et al. [52] tackle the enduring challenge of accuracy by introducing a novel hybrid approach. This method integrates three distinct models: an ARIMA model, a backpropagation neural network (BPNN) with adaptable parameters, and a fusion of ARIMA/ANN. This comprehensive strategy is applied to predict consumer price index (CPI) trends and cancer patient counts in Yemen's Ibb Province. Evaluation using diverse metrics underscores its effectiveness, with the ARIMA/BPNN and ARIMA/ANN models yielding significant improvements in accuracy and demonstrating a clear reduction in forecasting errors.

Commodity price forecasts can be categorized into two types, as described earlier, which are the neural network and hybridized neural network models. Even though hybrid models have been successful in forecasting the prices of various commodities. There has been no attempt to apply the following swarm intelligent techniques together as an optimizer in NN to predict the monthly iron ore prices: Harris Hawak, JAYA, Mouth Flame, and PSO. Moreover, two new techniques are proposed, which are the RWMFO and the GBMFO to improve the accuracy of the multilayer perceptron neural network prediction of the iron ore price.

## 3 Methodology

This section presents the MLP, the moth-flame optimization MFO algorithm used to boost the MLP to accurately predict the iron ore price, and the two proposed enhanced versions of MFO to be used by MLP. A thorough description is demonstrated to render a self-exploratory document. The stages of the proposed method are given in Figure 1.

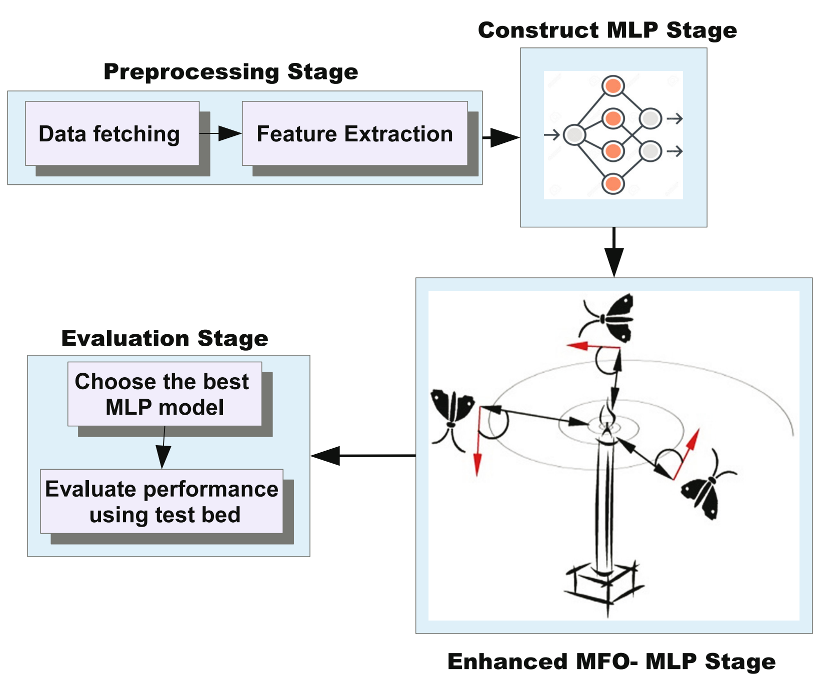



**Figure 1:** The stages of the proposed method.

### 3.1 Multilayer perceptron

Feedforward neural networks (FNNs) can be considered supervised learning mechanisms. They emulate the network architecture of the human brain wherein a set of neurons fragments into different layers. There is a direct connection between each layer and its followers. The neurons of FNN are designed to interconnect and group into three consecutive layers: input, hidden, and output. The number of neurons in the input layer reflects the number of features in the data. The hidden layers perform the necessary computation based on the model to map the input features to the targeted output in the output layers. In the output layer, the predicted class labels are represented by a set of output neurons FNN [53].

MLP is a popular variation of the FNN model. As FNN, MLP network architecture is organized by a set of interconnected neurons divided over input, hidden, and output layers. The main difference between FNN and MLP is that the connection in MLP has transferred in one way direction. An instance of MLP is visualized in the article [54], where three layers are given: input, single hidden, and output layers. The parameters of the MLP are the input data, the weights ($w$), and biases ($b$). The output of the MLP is computed based on these parameters as given in three steps:

*In the initial step*, in the architecture of the MLPs, each input is connected to a weighted sum score. This weighted sum is computed according to the equation provided in equation (1).

$$S_j = \sum_{i=1}^{n}(w_{ij}.\ X_i) - \beta_j, \quad j = 1, 2, \ldots, h, \tag{1}$$

where $n$ represents the overall count of inputs neurons, while $w_{ij}$ denotes the weight vector associated with the input neuron $i$ within the hidden neuron $j$. $X_i$ is the input number $i$ and $\beta_j$ is bias of hidden neuron $j$.

*In the second step*, typically, the output of the weighted vector is fed into the sigmoid activation function. Subsequently, the resulting output vector is forwarded to the next layer. However, in our proposed framework, we abstained from using the sigmoid activation function due to its output being constrained within the range of 0 and 1. Instead, we opted for the Leaky ReLU activation function, as suggested by Tsantekidis et al. [55]. The Leaky ReLU activation function generates an output that spans the range from negative infinity to positive infinity, making it more suitable for predicting iron ore prices (equation (2)).

Choosing the leaky rectified linear unit (Leaky ReLU) activation function over the sigmoid function for the MLP in predicting iron ore prices is advantageous due to leaky ReLU's ability to effectively model complex nonlinear relationships inherent in iron ore price determinants, and adapt to diverse feature influences. These characteristics enable the model to learn efficiently, converge faster, and capture intricate interactions among factors influencing iron ore prices, ultimately leading to improved predictive performance.

$$\text{Leaky ReLU} = \max(\alpha x, x). \tag{2}$$

It is important to note that the hyperparameter $\alpha$ typically falls within the interval of 0.01–0.1.

*In the final stage*, the calculation of the output in the last layer is performed as specified in equation (3).

$$\hat{y}_k = \sum_{i=1}^{m} w_{kj} f_i + b_k. \tag{3}$$

In the context of this discussion, $w_{jk}$ represents the weight connecting the hidden neuron $j$ to the output neuron $k$, while $b_k$ denotes the bias associated with output neuron $k$.

Significantly, the weight and bias vectors play a pivotal role in determining the final output within the MLP, as demonstrated by equations (1) and (3). Therefore, the quest for optimal values of these weight and bias vectors is a critical endeavor, as it directly contributes to enhancing the performance of the MLP model, leading to improved classification accuracy [56].

### 3.2 Moth-flame optimization algorithm based MLP (MFO-MLP)

A moth is an insect that is similar to butterflies [12]. There is a variety of species of this insect. The moth's life starts as larvae, which are transformed into a cocoon and then it becomes an adult. Interestingly, the moth has a special navigation technique at night called transverse orientation for navigation. During this navigation scheme, the moth keeps a steady angle to the moon. This ensures that the moth will fly in a straight line (Figure 2).

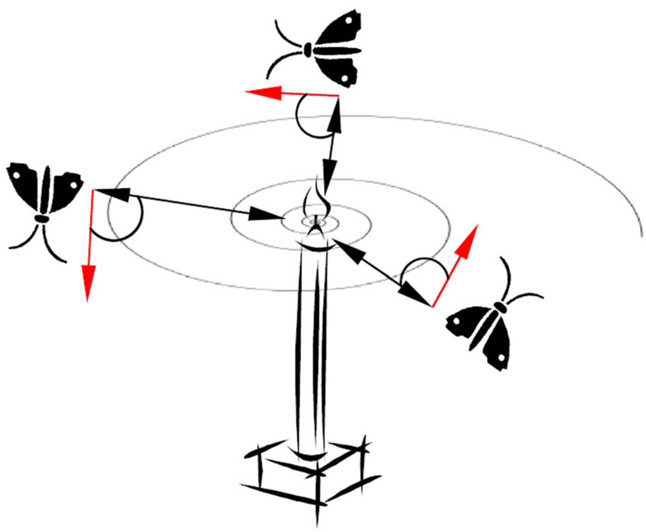


**Figure 2:** Moths spiral flying around a light source [12].

The MFO is a recently proposed swarm intelligence algorithm that is formulated to mimic the moth swarm behavior in nature [12]. The MFO starts by creating a set of random candidate solutions each one representing a moth. The problem variable values symbolize the moths' position. After that, each solution fitness is computed. The algorithm creates another set of flags called flames, which represent the moths with the best positions. This can help a moth not lose its best solution.

Each moth position is updated according to the flame using the following equation:

$$M_i = S(M_i, F_j). \tag{4}$$

Note that $M_i$ represents the $i$th moth, $F_j$ is the $j$th flame, and $S$ is a spiral function suggested by Mirjalili [12]. The spiral function utilized that simulates the spiral flying path of moths is formulated as follows:

$$M_i = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j. \tag{5}$$

As the equation shows, the next position of a moth is computed with respect to a flame. Note that $D_i$ represents how far the $i$th moth is from the $j$th flame, b is a constant to define the logarithmic spiral shape, and $t$ is a random number in $[-1, 1]$.

The following formula is used to compute $D$:

$$D_i = |F_j - M_i|. \tag{6}$$

Note that $M_i$ represents the $i$th moth, $F_j$ is the $j$th flame, and $D_i$ is how far the $i$th moth is from the $j$th flame.

To prevent MFO from being stuck in local optima the flames are sorted according to their fitness. After that, the moths modify their positions according to the relevant flames. The best flame is used to update the first moth position, while the worst flame is used to update the last moth position.

The order of the flames is modified according to the best solutions obtained so far in each evolution. The moths will change positions based on the changed flames. This would enhance the algorithm exploration. The algorithm reduces the number of flames during the progress of iterations using the following equation:

$$\text{flame no} = \text{round}\left(N - l \cdot \frac{N-1}{T}\right). \tag{7}$$

Note that $l$ is the current iteration number, $N$ is the highest possible number of flames, and $T$ is the algorithm's total iterations.

The proposed framework for predicting iron price is demonstrated in Figure 3. The proposed model is based on enhancing the performance of the neural network by selecting the most appropriate set of values of weights and biases using MFO.

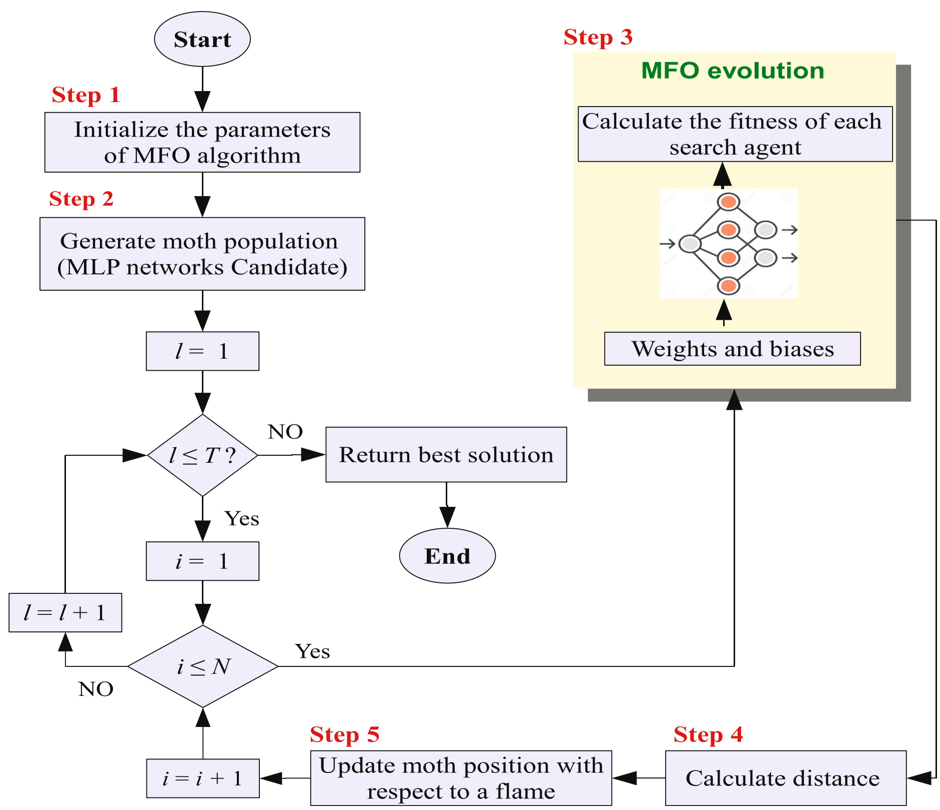



**Figure 3:** The steps of the proposed MFO-MLP.

## 3.3 Enhanced MFO-MLP

This section presents the two proposed enhanced MFO versions. The original MFO uses the logarithmic spiral shape of a random best solution from the population to develop a new solution. This can lead to not selecting the promising areas in the search space and thus being stuck in local optima. To improve MFO performance, the global best selection adapted from PSO is introduced to the algorithm. The new algorithm is called **GBMFO-NN**. It selects the best solution from the population to produce the new solution. Formally, if the decision variable is $x_i'$, then $x_i' = x_k^j$, where $j = \arg\min_{j \in [1,n]} f(\mathbf{x}^j) \wedge k \in (1, 2, \ldots, N)$.

Another version of MFO is introduced by applying the roulette wheel selection to improve the balance between algorithm exploration and exploitation. The new algorithm is called **RWMFO-NN**. This technique selects the solution $k$ from the population to choose the decision variable $x_i'$. Note that $x_i' = x_i^k$, where $k$ is picked randomly from $U(0, 1)$ using the roulette wheel method. Also, calculating the accumulative selection probability is done as follows: sum_prob = $\sum_{i=1}^{j} p_i$, where the selection probability $p_i$ for the solution $i$ is proportional to its fitness value, which is computed as follows: $p_i = \frac{f(x^i)}{\sum_{j=1}^{n} f(x^j)}$. The pseudo-code of the roulette wheel selection is shown in Algorithm 1.

**Algorithm 1**. The pseudo-code of roulette wheel selection method

1: Calculate the proportional probability for each solution for one island $P_j = \frac{f(x_j(l))}{\sum_{k=1}^{l_s} f(x_k(l))}$.
2: Set Sum_Prob = 0, Pindex = 0.
3: Generate $r \in [0, 1]$.
4: **while** (sum_prob ≤ $l$)
5: Pindex = Pindex + 1
6: sum_prob = sum_prob + $P_j$
7: **end while**
8: targeted = Pindex
9: Return targeted

The two new versions of MFO are used in MLP to obtain more promising weights and biases of the neural network to provide a more accurate prediction of iron ore prices.

# 4 Experiments and results

This section delves into the evaluation of the efficacy of the two novel approaches, RWMFO-NN and GBMFO-NN, in training MLP for accurate iron ore price prediction. The dataset is detailed in Section 4.1, while Section 4.2 shows the procedures for feature reduction and explores the correlation among the features. Section 4.3 is dedicated to presenting the performance metrics used in the evaluation. In Section 4.4, we analyze and evaluate the performance of the suggested RWMFO-NN and GBMFO-NN trainers in comparison to both the classical MFO-NN and several state-of-the-art algorithms.

All experiments were conducted in the Python programming language and executed on hardware with an Intel(R) Core(TM) i7-7700HQ CPU running at 2.80GHz, coupled with 16 GB of RAM.

## 4.1 Description of the datasets

The data utilized in this study encompass a 30-year span, spanning from January 1991 to December 2021, consisting of 360 months of iron ore price data. This dataset is divided into training and testing sets using two distinct ratios. Initially, a 70:30 ratio is employed, with 252 instances allocated for training the model and 108 instances dedicated to evaluating model performance. Subsequently, the dataset is divided into a 90:10 ratio, allocating 324 instances for training and 36 instances for testing.

The prediction model is constructed using a variety of financial indices sourced from different domains, each characterized by differing scales. Monthly data points are provided for all variables. To ensure consistent scaling, we apply the min–max normalization technique to each dataset, employing the following formula:

$$x' = \frac{x_i - X_{\min}}{X_{\max} - X_{\min}}. \quad (8)$$

Here, $x'$ represents the normalized value, while $X_{\min}$ and $X_{\max}$ denote the minimum and maximum values within each dataset.

To build a robust model for accurately estimating iron ore prices, we rely on the assessment of 12 specific parameters known to significantly influence iron ore prices, as outlined in Table 1. Our approach involved a meticulous analysis of a broader set of 574 parameters, assessing their impact on iron ore prices. For each parameter, we utilized a regression scoring function to quantify the strength of its linear relationship with the target variable. Parameters with higher scores are indicative of more robust and substantial correlations, thus providing insights into the statistical significance of these relationships.

**Table 1:** The datasets employed for predicting iron prices

| Features | Unit | Data source | Year of starting | Quantity of features |
|---|---|---|---|---|
| Iron and Copper | US Dollar/ton | World Bank; Thomson Reuters Datastream http://www.indexmundi.com/commodities/ | 1992 | 3 |
| price of silver | US Dollar/ozt | World Bank; Thomson Reuters Datastream http://www.indexmundi.com/commodities/ | 1992 | 3 |
| price of scrap | US Dollar/ton | U.S. Bureau of Labor Statistics https://fred.stlouisfed.org/series/PCU4299304299301 | 1980 | 3 |
| price of Oil | US Dollar/barrel | U.S. Energy Information Administration https://www.eia.gov/dnav/pet/pet_pri_spt_s1_m.htm | 1986 | 3 |
| price of gold | US Dollar/ozt | World Bank http://www.indexmundi.com/commodities/ | 1992 | 3 |
| Rates of exchange | — | Central banks http://fxtop.com/ | 1953 | 2 |
| Inflation rate - China | — | National Bureau of Statistics of China https://ieconomics.com/china-inflation-rate | 1988 | 2 |
| Inflation rate - US | — | U.S. Bureau of Labor Statistics https://inflationdata.com/Inflation/Inflation_Rate/Monthly_Inflation.aspx | 1913 | 2 |

Following the computation of scores, we employed a ranking procedure to identify the top ’k’ parameters exhibiting the highest scores. This selection process was guided by the principle that these primary parameters possess paramount predictive importance for the target variable within the scope of our investigation. After finalizing the parameter selection, we transformed the original feature matrix to exclusively encompass the chosen parameters. This refined feature matrix, comprising the selected parameters, serves as the foundation for subsequent modeling and comprehensive analysis. Our objective is to construct a robust model that delivers dependable estimates of iron ore prices, achieved by focusing on parameters with the most substantial influence.

## 4.2 Feature reductions using correlation analysis

The relationship between the iron ore prices and the features used for the prediction can be identified using correlation analysis. In this study, we developed two groups of input features using two techniques by analyzing correlations through Pearson’s correlation [1] (called *Set of attributes within Group A*) and categorizing the correlation within a proposed framework (called *Set of attributes within Group B*).

In the context of categorized correlation, features are organized into five distinct groups, namely, metal ore prices, inflation rates, oil prices, currency prices, and metal scrap prices. From this pool of features, the top eight features are chosen based on their strong correlation with the price of iron ore. The chosen features include:

- **Metal ore prices**: cooper price and silver price
- **Metal scrap prices**: commodity copper scrap and industry copper base scrap
- **Oil prices**: Europe brent spot price and crude oil
- **Currency prices**: USD/XAG and USD/CAD Table 2 demonstrates the pairwise correlation between the features with the highest correlation with the iron ore prices. Note that Group A of features represents the top nine features and Group B of features is the categorized correlation with eight features. Note that Pearson's correlation coefficient estimates the linear correlation degree among two variables. The values can be between +1 and −1. The +1 means complete correlation, 0 means no correlation, and −1 means negative correlation. Table 3 demonstrates the descriptive statistics of the selected features with the values mean, standard deviation (std), minimum (min), maximum (max), and coefficient of variation (CV).

## 4.3 Performance measures

The impact of the proposed model for forecasting the iron ore prices against other comparative models is conducted using several performance measures. This is achieved by assessing differences between the actual values and the forecast produced by the predication models as follows:

(1) Mean-square error (MSE): This metric is employed to determine the average squared disparity between the observed values and the forecasted values, as expressed in equation (9). Here, $D$ denotes the aggregate count of elements in the solution, whereas $\delta_j$ and $\hat{\delta}_j$ denote the actual and predicted values of the element at the $j$th position, respectively.

$$\text{MSE} = \frac{1}{D} \cdot \sum_{j=1}^{D} (\delta_j - \hat{\delta}_j)^2. \tag{9}$$

(2) RMSE: This measure calculates the square root of MSE as shown in equation (10).

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{D} (\delta_j - \hat{\delta}_j)^2}{D}}. \tag{10}$$

(3) Mean absolute error (MAE): This measure is calculated as the average absolute difference between the actual values and the predicted values as shown in equation (11).

$$\text{MSE} = \frac{1}{D} \cdot \sum_{j=1}^{D} |\delta_j - \hat{\delta}_j|. \tag{11}$$

(4) Average absolute relative deviation (AARD): It is used to calculate the percentage of the difference between the actual value and the predicted value as shown in equation (12).

$$\text{AARD} = \frac{100}{D} \cdot \sum_{j=1}^{D} \left| \frac{\delta_j - \hat{\delta}_j}{\delta_j} \right|. \tag{12}$$

## 4.4 Comparison results

In this section, the efficiency of using six metaheuristic trainers to predict iron ore prices is studied. Table 4 illustrates the abbreviations and the full names of the comparative algorithms. It should be noted that all these comparative algorithms run under the same parameter settings to make fair comparisons. These parameter settings are set as follows: the population size is 30, the maximum number of iterations is equal to 100, and the weight and bias receive values within the range of −1 to 1. Each algorithm executes 30 times to study its robustness. In addition, the FNN has a hidden layer with seven neurons.

Table 5 presents the statistical outcomes of various comparative algorithms, evaluating their performance in relation to MSE, RMSE, MAE, and AARD when applied to two distinct feature groups. Details of these feature

**Table 2:** Correlation matrix

| Group A of features | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Feature** | **Iron ore price** | **Copper price** | **Silver price** | **Gold price** | **Scrap** | **China inflation change** | **US inflation change** | **USD/AUD** | **USD/CNY** | **Crude oil** |
| Iron ore price | 1.0000 | | | | | | | | | |
| Copper price | 0.8934 | 1.0000 | | | | | | | | |
| Silver price | 0.8447 | 0.8942 | 1.0000 | | | | | | | |
| Gold price | 0.7660 | 0.8614 | 0.9226 | 1.0000 | | | | | | |
| Scrap | 0.8894 | 0.9383 | 0.8695 | 0.8324 | 1.0000 | | | | | |
| China inflation change | 0.2070 | 0.2658 | 0.2559 | 0.2598 | 0.2612 | 1.0000 | | | | |
| US inflation change | 0.1964 | 0.1690 | 0.1058 | 0.0693 | 0.1841 | 0.0348 | 1.0000 | | | |
| USD/AUD | 0.6673 | 0.7307 | 0.6909 | 0.5769 | 0.7343 | 0.0867 | 0.0634 | 1.0000 | | |
| USD/CNY | 0.5090 | 0.5843 | 0.6092 | 0.6700 | 0.5606 | 0.3062 | 0.0332 | 0.4985 | 1.0000 | |
| Crude oil | 0.8555 | 0.8744 | 0.7944 | 0.6873 | 0.8897 | 0.2970 | 0.2053 | 0.7189 | 0.5070 | 1.0000 |

| Group B of features | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Feature** | **Iron ore** | **Commodity copper scrap** | **Copper price** | **Industry copper base scrap** | **Europe brent spot price** | **Crude oil** | **Silver price** | **USD/XAG** | **USD/CAD** |
| Iron ore | 1 | | | | | | | | |
| Commodity copper scrap | 0.8977 | 1 | | | | | | | |
| Copper price | 0.8934 | 0.9952 | 1 | | | | | | |
| Industry Copper Base Scrap | 0.8926 | 0.9945 | 0.9906 | 1 | | | | | |
| Europe brent spot price | 0.8562 | 0.8978 | 0.8854 | 0.9100 | 1 | | | | |
| Crude oil | 0.8555 | 0.8872 | 0.8744 | 0.8938 | 0.9897 | 1 | | | |
| Silver price | 0.8447 | 0.8956 | 0.8942 | 0.9154 | 0.8465 | 0.7944 | 1 | | |
| USD/XAG | 0.7960 | 0.9120 | 0.9184 | 0.9295 | 0.8308 | 0.8148 | 0.8820 | 1 | |
| USD/CAD | 0.7662 | 0.8173 | 0.7890 | 0.8046 | 0.8342 | 0.8409 | 0.7000 | 0.6885 | 1 |

**Table 3:** Summary statistics for input attributes

| Attribute | N | Mean | Std | Min | Max | CV (%) |
|---|---|---|---|---|---|---|
| **Set of attributes within group A** | | | | | | |
| Iron ore | 360 | 72.60169916 | 49.37414678 | 26.47 | 214.43 | 68.00687498 |
| Copper price | 360 | 4638.246267 | 2569.275395 | 1377.28 | 10161.97 | 55.39325095 |
| Gold price | 360 | 831.534429 | 523.7065688 | 256.08 | 1968.63 | 62.98074386 |
| Silver price | 360 | 12.77231198 | 8.708394618 | 3.65 | 42.7 | 68.18181887 |
| Scrap | 360 | 305.7488384 | 154.8948672 | 101.2 | 731.4 | 50.6608195 |
| Crude oil | 360 | 49.7208078 | 28.81300393 | 11.35 | 133.88 | 57.94958934 |
| USA/CNY | 360 | 7.295243384 | 0.946118195 | 5.440707 | 8.726455 | 12.96897369 |
| USA/AUD | 360 | 1.358296209 | 0.232089964 | 0.928882 | 2.001456 | 17.0868447 |
| China inflation change | 360 | 0.006195522 | 0.005759249 | 0 | 0.0405 | 92.95824507 |
| US inflation change | 360 | 0.003001393 | 0.002427571 | 0 | 0.0192 | 80.88146779 |
| **Set of attributes within Group B** | | | | | | |
| Iron ore | 360 | 72.60169916 | 49.37414678 | 26.47 | 214.43 | 68.00687498 |
| Commodity copper scrap | 360 | 278.3465571 | 148.2167749 | 87.1 | 595.7 | 53.24900599 |
| Copper price | 360 | 4638.246267 | 2569.275395 | 1377.28 | 10161.97 | 55.39325095 |
| Industry copper base scrap | 360 | 347.2198357 | 194.1495818 | 105.3 | 719.5 | 55.91546389 |
| Europe brent spot price | 360 | 51.37367688 | 32.37235313 | 9.82 | 132.72 | 63.01350244 |
| Crude oil | 360 | 49.7208078 | 28.81300393 | 11.35 | 133.88 | 57.94958934 |
| Silver price | 360 | 12.77231198 | 8.708394618 | 3.65 | 42.7 | 68.18181887 |
| USD/XAG | 360 | 4.007307958 | 2.415128936 | 0.8203 | 8.786158 | 60.26811419 |
| USD/CAD | 360 | 1.255133084 | 0.168514815 | 0.907078 | 1.58816 | 13.42605159 |

**Table 4:** The comparative algorithms

| Abb. | Model name |
|---|---|
| HHO-NN | Harris hawks optimization using a feedforward NN |
| JAYA-NN | JAYA algorithm using a feedforward NN |
| PSO-NN | Particle swarm optimization using a feedforward NN |
| MFO-NN | moth-flame optimization algorithm using a feedforward NN |
| GBMFO-NN | Global-best moth-flame optimization algorithm using a feedforward NN |
| RWMFO-NN | Roulette wheel moth-flame optimization algorithm using a feedforward NN |

groups can be found in Table 2. The table reveals the results from two separate experiments: one where the dataset was partitioned with a 70% allocation for training and 30% for testing, and another with a 90% training and 10% testing split. It is worth noting that lower values of MSE, RMSE, MAE, and AARD indicate superior performance, and the best results are highlighted in bold.

Considering Group A of features, the proposed RWMFO-NN stands out as the top performer, achieving the best results for MSE, RMSE, and MAE in the 70:30 ratio scenario. In terms of AARD in the same ratio, the RWMFO-NN ranks fourth. The proposed MFO-NN secures the second position by delivering the second-lowest scores for MSE, RMSE, MAE, and AARD in the 70:30 scenario. The proposed GBMFO-NN takes the third spot, with the third lowest MSE, RMSE, and AARD scores, although it ranks fourth in MAE.

In the 90:10 ratio scenario, the proposed MFO-NN takes the lead by achieving the best results for MSE, RMSE, and MAE, while it ranks third in AARD. The RWMFO-NN follows in second place, producing the second-best MAE and AARD results, and the third-best MSE and RMSE results in the 90:10 scenario. Meanwhile, the proposed GBMFO-NN secures the fourth position across all metrics in the 90:10 scenario.

According to Group B of features, it can be seen that the proposed MFO-NN is ranked first by obtaining the best results in all measures for the ratio 70:30. Furthermore, the proposed GBMFO-NNN is ranked second in MSE, RMSE, and MAE. While it is ranked third in AARD for the 70:30 ratio. The proposed RWMFO-NN is ranked third in

**Table 5:** Performance comparison through out-of-sample testing

| Algorithm | Splitting the dataset into 70:30 | | | | Splitting the dataset into 90:10 | | | |
|---|---|---|---|---|---|---|---|---|
| | MSE | RMSE | MAE | AARD | MSE | RMSE | MAE | AARD |
| **Set of attributes within Group A** | | | | | | | | |
| HHO-NN | 0.0116 | 0.1075 | 0.0731 | **51.6766** | 0.0110 | 0.1047 | 0.0717 | **52.6268** |
| JAYA-NN | 0.0261 | 0.1617 | 0.1066 | 64.6785 | 0.0241 | 0.1552 | 0.1041 | 63.3580 |
| PSO-NN | 0.0141 | 0.1186 | 0.0787 | 55.0351 | 0.0127 | 0.1125 | 0.0765 | 55.9429 |
| MFO-NN | 0.0101 | 0.1007 | 0.0708 | 53.8428 | **0.0106** | **0.1031** | **0.0688** | 54.3443 |
| GBMFO-NN | 0.0110 | 0.1049 | 0.0733 | 55.3345 | 0.0123 | 0.1110 | 0.0731 | 55.7267 |
| RWMFO-NN | **0.0098** | **0.0988** | **0.0693** | 55.9058 | 0.0119 | 0.1092 | 0.0706 | 53.5232 |
| **Set of attributes within Group B** | | | | | | | | |
| HHO-NN | 0.0113 | 0.1062 | 0.0738 | 51.1438 | 0.0112 | 0.1057 | 0.0685 | 49.7211 |
| JAYA-NN | 0.0155 | 0.1245 | 0.0857 | 55.7095 | 0.0146 | 0.1207 | 0.0858 | 56.5713 |
| PSO-NN | 0.0096 | 0.0977 | 0.0684 | 56.8330 | 0.0108 | 0.1041 | 0.0660 | 53.6876 |
| MFO-NN | **0.0092** | **0.0957** | **0.0613** | **50.3819** | 0.0080 | 0.0893 | 0.0555 | 51.0557 |
| GBMFO-NN | 0.0093 | 0.0965 | 0.0645 | 52.5321 | 0.0093 | 0.0966 | 0.0608 | **49.0405** |
| RWMFO-NN | 0.0107 | 0.1032 | 0.0655 | 52.9559 | **0.0075** | **0.0866** | **0.0526** | 50.1753 |

Bold values are the algorithm that obtained the best results in each one of the metrics.

MSE and MAE. While it is ranked fourth in RMSE and AARD for the ratio of 70:30. On the other hand, the proposed RWMFO-NN is ranked first by obtaining the best results in MSE, RMSE, and MAE. While achieving the second-highest outcomes in AARD pertaining to the 90:10 ratio. The proposed MFO-NN is ranked second by achieving the second-best results in MSE, RMSE, and MAE. While it obtained the third-best results in AARD for the ratio 90:10. The proposed GBMFO-NN is ranked first by obtaining the best AARD results. While it achieved the third-best results in MSE, RMSE, and MAE. Table 5 shows that the proposed models achieve better results when the second group of features (Group B) is used as input when compared to the first group of features (Group A). This proves that the set of attributes within Group B are more promising to be used to predict iron ore prices.

The performance of the suggested MFO-NN, GBMFO-NN, and RWMFO-NN models was compared against the other models during the training process as shown in Figures 4 and 5. The $x$-axis represents the time in months, while the $y$-axis is the iron ore prices. It can be clearly observed that the proposed models are able to accurately predict the iron ore price in all cases of experiments. However, it is evident that that the JAYA-NN model fails to predict accurately the iron ore prices for all cases of experiments.

In addendum to the above, the performance of the proposed models (i.e., MFO-NN, GBMFO-NN, and RWMFO-NN) against the other comparative models on the test data are illustrated in Figures 6 and 7. From these figures, we can observe that all models can effectively predict the iron ore price in all cases of experiments.

For more validation, the performance of the proposed models is compared with the classical machine learning training models as depicted in Table 6. The traditional trainers are Huber regressor (HR), linear regression (LR), MLP regressor (MLP), random sample consensus (RANSAC) regressor, and TheilSen regressor. The results of the proposed models are compared with the classical training models in two phases as follows: the first one, when the data are divided into 70:30, and the second one when the data are divided into 90:10. Table 6 shows that the proposed algorithms perform better than the comparable classical models concerning MSE, RMSE, and MAE when the methods are applied to the first group of features (Group A). On the other hand, the performance of the proposed algorithms is better than the other classical models in all measures when the methods are applied to the second group of features (Group B).

The enhanced versions of the MFO, namely, GBMFO-NN and RWMFO-NN, are necessary to improve the training process of the MLP for iron ore price prediction. This is important due to the challenges in selecting optimal weights and biases for the MLP. The proposed GBMFO-NN and RWMFO-NN algorithms utilize moths' navigation strategies and incorporate novel selection methods. These enhancements address the limitations of traditional optimization methods, resulting in improved convergence and avoidance of local optima. Applied

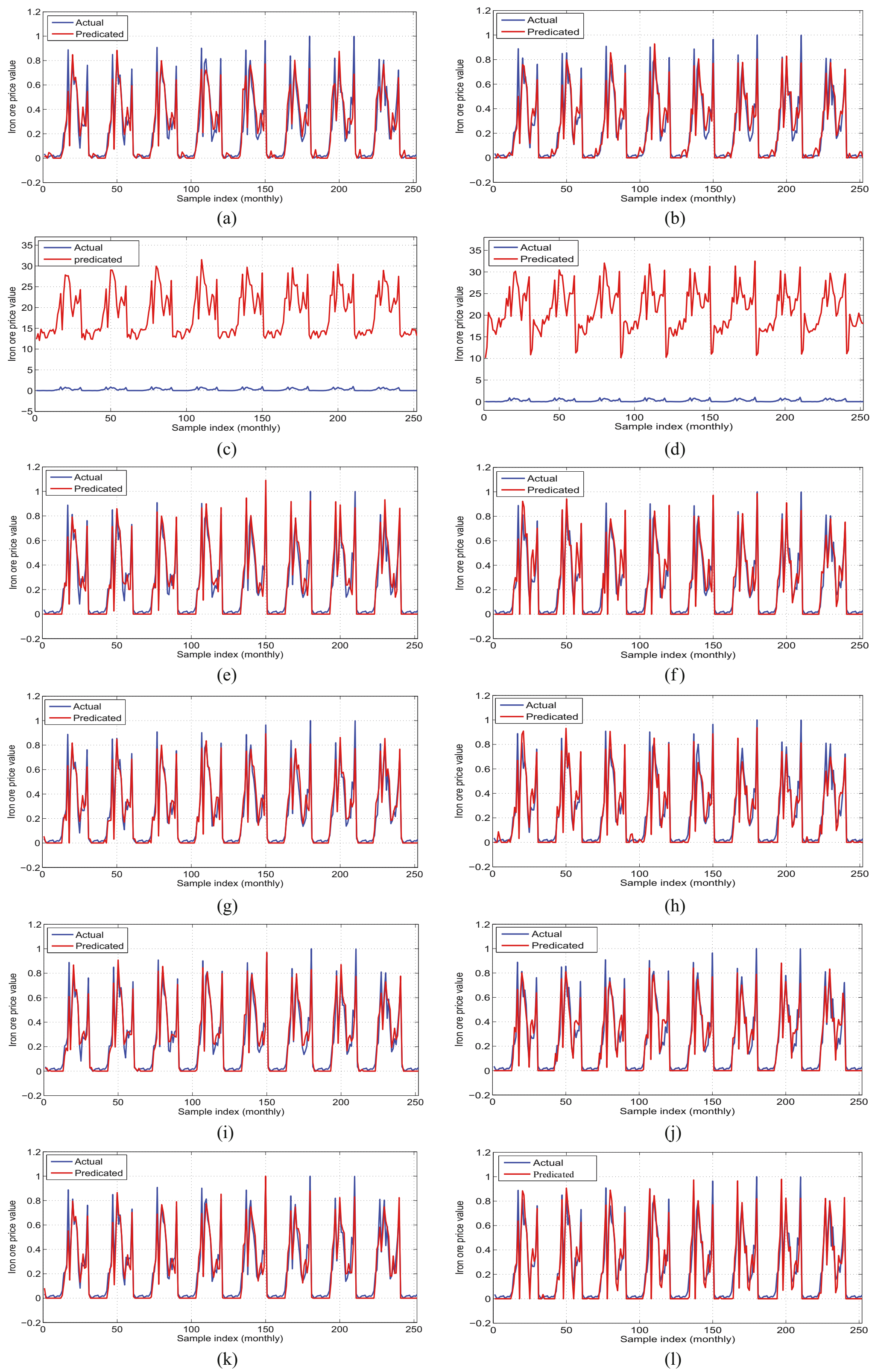



**Figure 4:** The performance of all models on the training set split into a 70:30 ratio. (a) HHO-NN (Group A), (b) HHO-NN (Group B), (c) JAYA-NN (Group A), (d) JAYA-NN (Group B), (e) PSO-NN (Group A), (f) PSO-NN (Group B), (g) MFO-NN (Group A), (h) MFO-NN (Group B), (i) GBMFO-NN (Group A), (j) GBMFO-NN (Group B), (k) RWMFO-NN (Group A), and (l) RWMFO-NN (Group B).

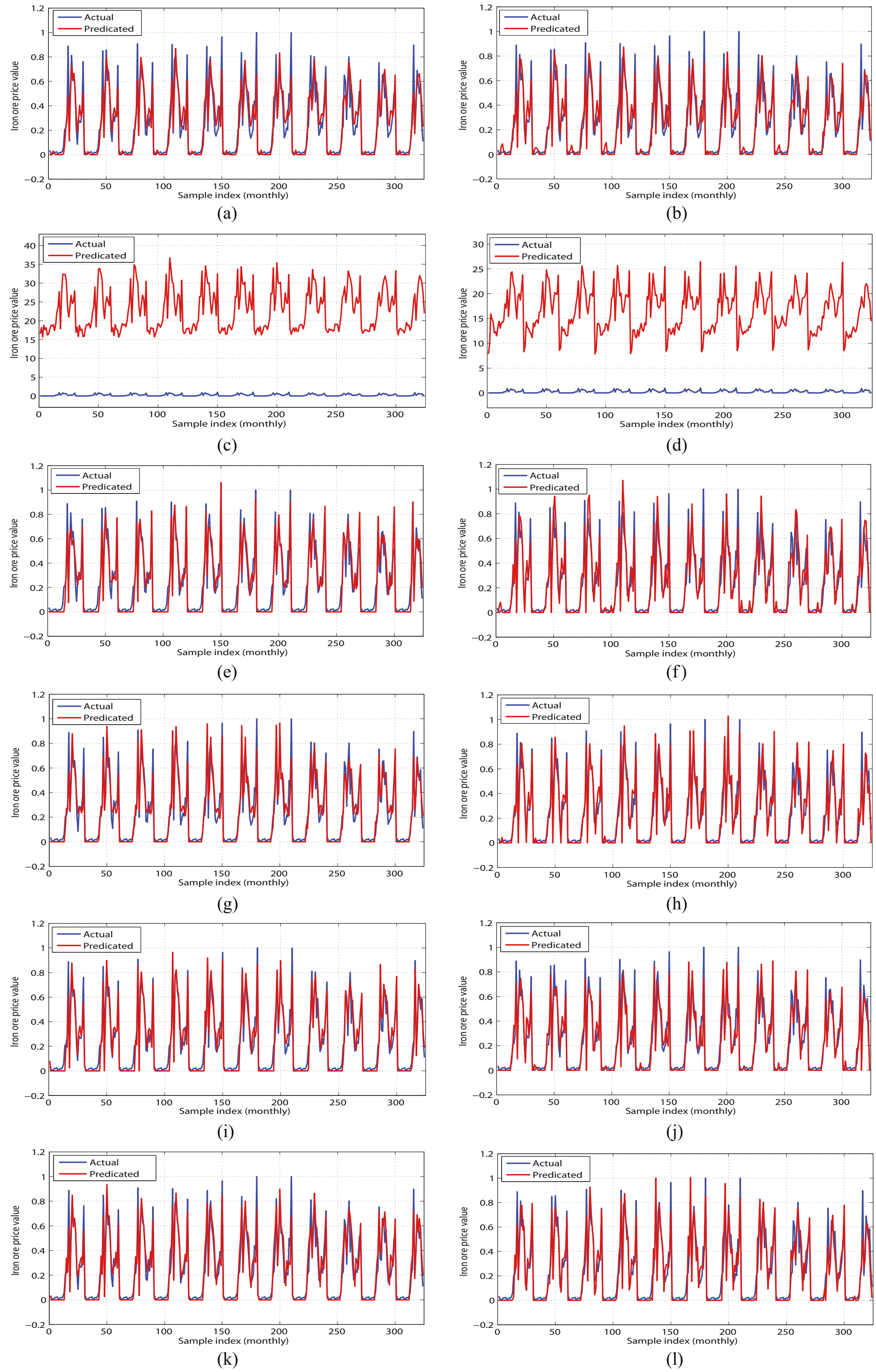
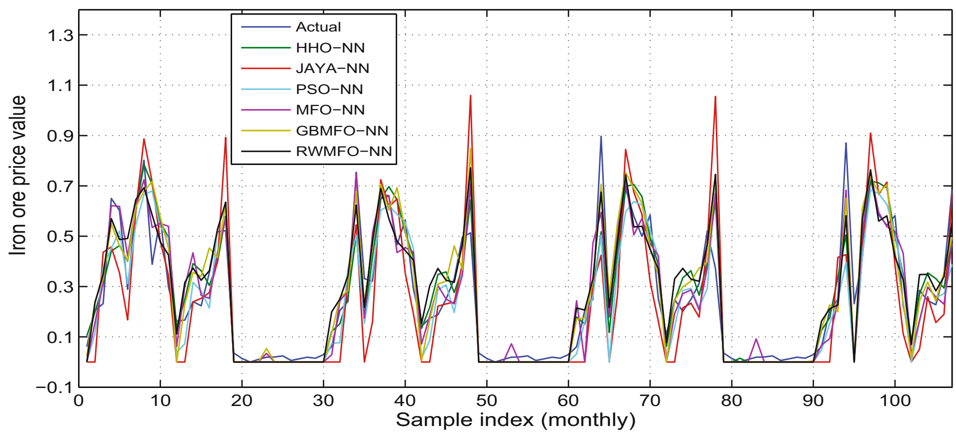
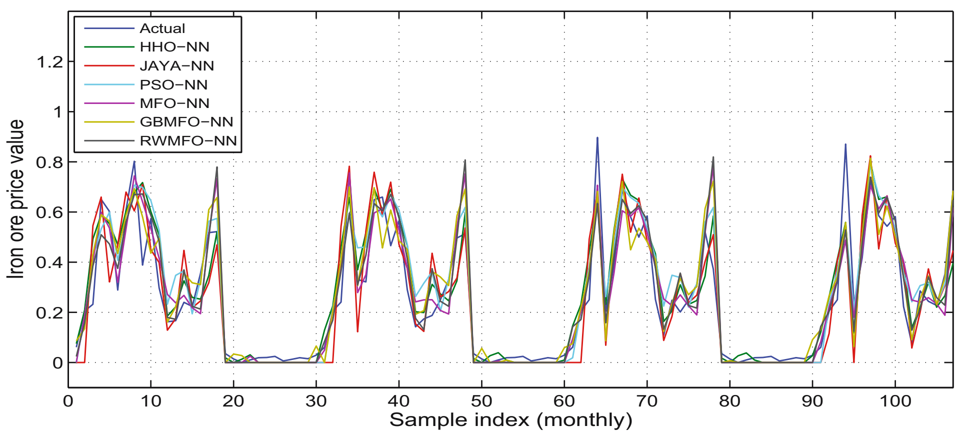



**Figure 5:** The performance of all models on the training set split into a 90:10 ratio. (a) HHO-NN (Group A), (b) HHO-NN (Group B), (c) JAYA-NN (Group A), (d) JAYA-NN (Group B), (e) PSO-NN (Group A), (f) PSO-NN (Group B), (g) MFO-NN (Group A), (h) MFO-NN (Group B), (i) GBMFO-NN (Group A), (j) GBMFO-NN (Group B), (k) RWMFO-NN (Group A), and (l) RWMFO-NN (Group B).




(a)




(b)

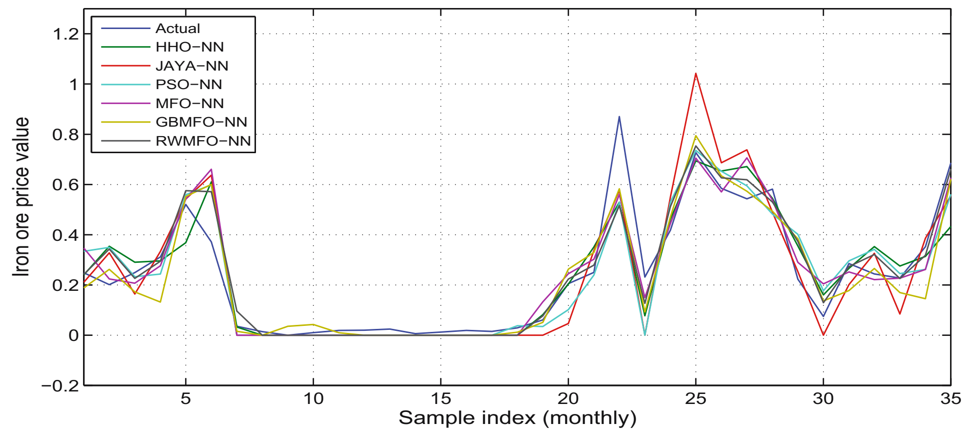
**Figure 6:** The performance of all models on the testing set split into a 70:30 ratio. (a) Group A of features and (b) Group B of features.

to the context of iron ore price prediction, these versions demonstrate better accuracy, as shown by comparisons with various other optimization algorithms. The integration of these enhanced MFO algorithms with the MLP neural network presents a promising approach for accurately forecasting iron ore prices.

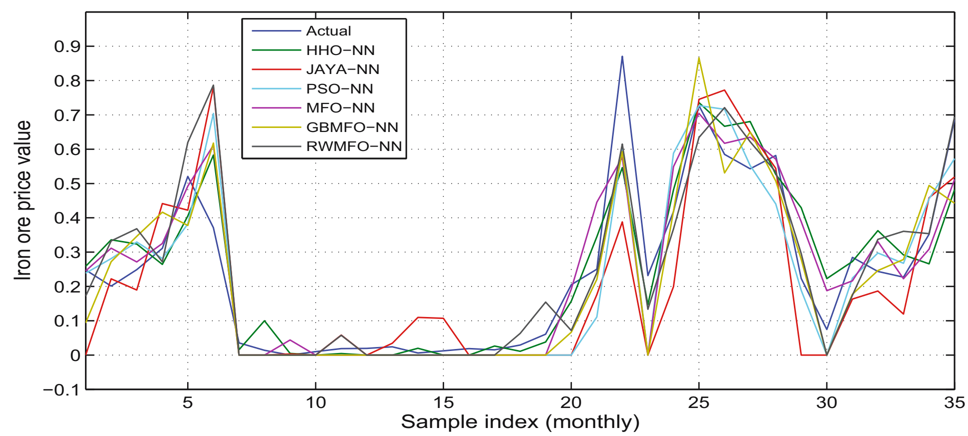
## 5 Conclusion and future work

This article proposes a new model to forecast iron ore prices based on two enhanced MFOs by novel selection methods for training the FNN called MLP. The main purpose of the proposed model is to help in predicting the iron ore price to help stakeholders by supporting their decision-making and reducing the market risk. Initially, the data are extracted and feature reduction is applied based on correlation analysis. Thereafter, the performance of MFO is improved by integrating roulette wheel selection to propose RWMFO-NN. In addition, the global-best selection method is adapted to propose GBMFO-NN. The two proposed versions of MFO serve as trainers for MLP. The proposed model's outcomes are compared with six optimizers used as trainers for MLP: the original MFO (MFO-NN), JAYA algorithm (JAYA-NN), particle swarm optimization (PSO-NN), and Harris Hawks optimizer (HHO-NN).

We employed eight datasets encompassing metal ore prices, inflation rates, oil prices, currency prices, and metal scrap prices as input for our framework. These datasets were subjected to min–max normalization. We evaluated two distinct sets of features: the first (group A) was selected based on Pearson's correlation [1], while




(a)




(b)

**Figure 7:** The performance of all models on the testing set split into a 90:10 ratio. (a) Group A of features and (b) Group B of features.

**Table 6:** Comparison of out-of-sample testing performance against conventional models

| Algorithm | Splitting the dataset into 70:30 | | | | Splitting the dataset into 90:10 | | | |
|---|---|---|---|---|---|---|---|---|
| | MSE | RMSE | MAE | AARD | MSE | RMSE | MAE | AARD |
| **Set of attributes within Group A** | | | | | | | | |
| HR | 0.0142 | 0.1190 | 0.0818 | 67.3086 | 0.0146 | 0.1210 | 0.0792 | 59.5655 |
| LR | 0.0151 | 0.1227 | 0.0861 | 92.4681 | 0.0164 | 0.1279 | 0.0866 | 87.1033 |
| MLP | 0.0111 | 0.1051 | 0.0706 | **50.7207** | 0.0128 | 0.1131 | 0.0744 | **48.6672** |
| RANSAC | 0.0150 | 0.1224 | 0.0836 | 72.0497 | 0.0146 | 0.1208 | 0.0769 | 54.2607 |
| TSR | 0.0126 | 0.1120 | 0.0794 | 81.0395 | 0.0136 | 0.1165 | 0.0790 | 79.0136 |
| MFO-NN | 0.0101 | 0.1007 | 0.0708 | 53.8428 | **0.0106** | **0.1031** | **0.0688** | 54.3443 |
| GBMFO-NN | 0.0110 | 0.1049 | 0.0733 | 55.3345 | 0.0123 | 0.1110 | 0.0731 | 55.7267 |
| RWMFO-NN | **0.0098** | **0.0988** | **0.0693** | 55.9058 | 0.0119 | 0.1092 | 0.0706 | 53.5232 |
| **Set of attributes within Group B** | | | | | | | | |
| HR | 0.0146 | 0.1210 | 0.0792 | 59.5655 | 0.0130 | 0.1142 | 0.0729 | 72.0382 |
| LR | 0.0164 | 0.1279 | 0.0866 | 87.1033 | 0.0137 | 0.1168 | 0.0778 | 82.1880 |
| MLP | 0.0136 | 0.1165 | 0.0783 | 70.1665 | 0.0109 | 0.1044 | 0.0691 | 60.2139 |
| RANSAC | 0.0148 | 0.1216 | 0.0779 | 53.8818 | 0.0127 | 0.1126 | 0.0696 | 68.5953 |
| TSR | 0.0136 | 0.1168 | 0.0787 | 76.9026 | 0.0121 | 0.1101 | 0.0668 | 55.0670 |
| MFO-NN | **0.0092** | **0.0957** | **0.0613** | **50.3819** | 0.0080 | 0.0893 | 0.0555 | 51.0557 |
| GBMFO-NN | 0.0093 | 0.0965 | 0.0645 | 52.5321 | 0.0093 | 0.0966 | 0.0608 | **49.0405** |
| RWMFO-NN | 0.0107 | 0.1032 | 0.0655 | 52.9559 | **0.0075** | **0.0866** | **0.0526** | 50.1753 |

Bold values are the algorithm that obtained the best results in each one of the metrics.

the second (group B) was chosen using a novel categorized Pearson's correlation approach. Both groups served as inputs for the GBMFO-NN and RWMFO-NN models, each subjected to two training/testing scenarios: 70:30 and 90:10.

The results indicate that the prediction of iron ore prices is more accurate when employing group B features. Our findings demonstrate that the proposed MFO-NN, GBMFO-NN, and RWMFO-NN methods outperform other comparative approaches in terms of RMSE, MSE, AARD, and MAE. In summary, the MFO, GBMFO-NN, and RWMFO-NN models stand out as promising methods for accurately forecasting iron ore prices.

While the proposed research offers promising advancements in predicting iron ore prices using enhanced optimization algorithms, there are several limitations and potential biases that should be acknowledged. The effectiveness of the proposed model relies on the assumption that the selected input features (based on Pearson's correlation and categorized correlation) accurately capture the underlying relationships influencing iron ore prices, yet the process of feature selection can introduce its own biases and assumptions, potentially overlooking important variables or inadvertently incorporating noise.

The study also assumes that the MFO algorithms, as applied to enhance the training of the MLP, are universally beneficial across diverse datasets, which might not hold true in all scenarios. Furthermore, the normalization technique employed, specifically min–max normalization, assumes that the distribution of input data are uniform and neglects potential outliers or skewed distributions that could impact model performance. In addition, while the proposed models demonstrate improved prediction accuracy, the generalization of these results to different time periods or economic contexts requires cautious consideration due to the potential nonstationarity and evolving dynamics of economic factors affecting iron ore prices. Overall, while the study makes valuable contributions to the field of iron ore price prediction, the highlighted limitations and assumptions should be kept in mind when interpreting the implications and generalizability of the findings.

As the MFO-NN and its two variants show very successful prediction models for iron ore prices, in the future, other features can be added to the models for enhancing the forecasting accuracy. Also, the proposed models can be implemented for other data such as forecasting gold or silver prices.

**Funding information:** The authors state no funding involved.

**Author contributions:** Iyad Abu Doush, Basem Ahmed, Mohammed A. Awadallah develop the conceptualization, Methodology, and Software. Mohammed A. Awadallah and Mohammed Azmi Al-Betar did the validation and experiements. Noor Aldeen Alawad did formal analysis and visualization. All authors reviewed the manuscript.

**Conflict of interest**: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this research.

**Data availability statement:** The data that support the findings of this study are available from the corresponding author upon reasonable request.

## References

[1] Ewees AA, Abd Elaziz M, Alameer Z, Ye H, Jianhua Z. Improving multilayer perceptron neural network using chaotic grasshopper optimization algorithm to forecast iron ore price volatility. Resources Policy. 2020;65:101555.
[2] Wu J, Yang J, Ma L, Li Z, Shen X. A system analysis of the development strategy of iron ore in China. Resources Policy. 2016;48:32–40.
[3] Ma W, Zhu X, Wang M. Forecasting iron ore import and consumption of China using grey model optimized by particle swarm optimization algorithm. Resources Policy. 2013;38(4):613–20.
[4] Li D, Moghaddam MR, Monjezi M, Jahed Armaghani D, Mehrdanesh A. Development of a group method of data handling technique to forecast iron ore price. Appl Sci. 2020;10(7):2364.
[5] Malanichev A, Vorobyev P. Forecast of global steel prices. Studies Russian Economic Development. 2011;22(3):304–11.
[6] Hafezi R, Akhavan A. Forecasting gold price changes: Application of an equipped artificial neural network. AUT J Model Simulat. 2018;50(1):71–82.

[7] Al-Betar MA, Doush IA, Khader AT, Awadallah MA. Novel selection schemes for harmony search. Appl Math Comput. 2012;218(10):6095–117.

[8] Zubaidi SL, Abdulkareem IH, Hashim KS, Al-Bugharbee H, Ridha HM, Gharghan SK, et al. Hybridised artificial neural network model with slime mould algorithm: a novel methodology for prediction of urban stochastic water demand. Water. 2020;12(10):2692.

[9] Zhang H, Nguyen H, Bui XN, Pradhan B, Mai NL, Vu DA. Proposing two novel hybrid intelligence models for forecasting copper price based on extreme learning machine and meta-heuristic algorithms. Resources Policy. 2021;73:102195.

[10] Alameer Z, Abd Elaziz M, Ewees AA, Ye H, Jianhua Z. Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm. Resources Policy. 2019;61:250–60.

[11] Doush IA, Sawalha A. Automatic music composition using genetic algorithm and artificial neural networks. Malaysian J Comput Sci. 2020;33(1):35–51.

[12] Mirjalili S. moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-based Syst. 2015;89:228–49.

[13] Sahoo SK, Saha AK, Ezugwu AE, Agushaka JO, Abuhaija B, Alsoud AR, et al. Moth flame optimization: theory, modifications, hybridizations, and applications. Archives Comput Methods Eng. 2023;30(1):391–426.

[14] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: Algorithm and applications. Future Generation Comput Syst. 2019;97:849–72.

[15] Rao R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Industr Eng Comput. 2016;7(1):19–34.

[16] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. vol. 4. IEEE; 1995. p. 1942–8.

[17] Morariu N, Iancu E, Vlad S. A neural network model for time series forecasting. Romanian J Econ Forecast. 2009;4:213–23.

[18] Lineesh M, Minu K, John CJ. Analysis of nonstationary nonlinear economic time series of gold price: a comparative study. In: International Mathematical Forum. vol. 5. Citeseer; 2010. p. 1673–83.

[19] Parisi A, Parisi F, Díaz D. Forecasting gold price changes: Rolling and recursive neural network models. J Multinational Financial Management. 2008;18(5):477–87.

[20] Kohzadi N, Boyd MS, Kermanshahi B, Kaastra I. A comparison of artificial neural network and time series models for forecasting commodity prices. Neurocomputing. 1996;10(2):169–81.

[21] Kristjanpoller W, Fadic A, Minutolo MC. Volatility forecast using hybrid neural network models. Expert Syst Appl. 2014;41(5):2437–42.

[22] Fath AH, Madanifar F, Abbasi M. Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems. Petroleum. 2020;6(1):80–91.

[23] Ramyar S, Kianfar F. Forecasting crude oil prices: A comparison between artificial neural networks and vector autoregressive models. Comput Econ. 2019;53(2):743–61.

[24] Kim KJ. Artificial neural networks with evolutionary instance selection for financial forecasting. Expert Syst Appl. 2006;30(3):519–26.

[25] Bildirici M, Ersin ÖÖ. Forecasting oil prices: Smooth transition and neural network augmented GARCH family models. J Petroleum Sci Eng. 2013;109:230–40.

[26] Livieris IE, Pintelas E, Pintelas P. A CNN-LSTM model for gold price time-series forecasting. Neural Comput Appl. 2020;32(23):17351–60.

[27] Primananda SB, Isa SM. Forecasting Gold price in Rupiah using multivariate analysis with LSTM and GRU neural networks. Adv Sci Tech Eng Syst J. 2021;6(2):245–53.

[28] Varma GN, Padma K. Forecasting agricultural commodity pricing using neural network-based approach. Int J Business Inform Syst. 2019;31(4):517–29.

[29] Xu X, Zhang Y. Thermal coal price forecasting via the neural network. Intelligent Syst Appl. 2022;14:200084.

[30] Xu X, Zhang Y. Canola and soybean oil price forecasts via neural networks. Adv Comput Intell. 2022;2(5):32.

[31] Ding H. Prediction of retail price of sporting goods based on lstm network. Comput Intell Neurosci. 2022;2022.

[32] Xu X, Zhang Y. Regional steel price index forecasts with neural networks: evidence from East, South, North, Central South, Northeast, Southwest, and Northwest China. J Supercomput. 2023;79:1–19.

[33] Joshi D, Chithaluru P, Anand D, Hajjej F, Aggarwal K, Torres VY, et al. An evolutionary technique for building neural network models for predicting metal prices. Mathematics. 2023;11(7):1675.

[34] Khashei M, Bijari M. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Appl Soft Comput. 2011;11(2):2664–75.

[35] Althelaya KA, El-Alfy ESM, Mohammed S. Evaluation of bidirectional LSTM for short-and long-term stock market prediction. In: 2018 9th International Conference on Information and Communication Systems (ICICS). IEEE; 2018. p. 151–6.

[36] Naik N, Mohan BR. Study of stock return predictions using recurrent neural networks with LSTM. In: International Conference on Engineering Aapplications of Neural Networks. Springer; 2019. p. 453–9.

[37] Cheng LC, Huang YH, Wu ME. Applied attention-based LSTM neural networks in stock prediction. In: 2018 IEEE International Conference on Big Data (Big Data). IEEE; 2018. p. 4716–8.

[38] Alameer Z, Fathalla A, Li K, Ye H, Jianhua Z. Multistep-ahead forecasting of coal prices using a hybrid deep learning model. Resources Policy. 2020;65:101588.

[39] Baffour AA, Feng J, Taylor EK. A hybrid artificial neural network-GJR modeling approach to forecasting currency exchange rate volatility. Neurocomputing. 2019;365:285–301.

[40] García D, Kristjanpoller W. An adaptive forecasting approach for copper price volatility through hybrid and non-hybrid models. Appl Soft Comput. 2019;74:466–78.
[41] Nguyen-Ky T, Mushtaq S, Loch A, Reardon-Smith K, An-Vo DA, Ngo-Cong D, et al. Predicting water allocation trade prices using a hybrid Artificial Neural Network-Bayesian modelling approach. J Hydrol. 2018;567:781–91.
[42] Kristjanpoller W, Minutolo MC. A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis. Expert Syst Appl. 2018;109:1–11.
[43] Jannah M, Sadik K, Afendi FM. Study of forecasting method for agricultural products using hybrid ANN-GARCH approach. In: Journal of Physics: Conference Series. vol. 1863. IOP Publishing; 2021. p. 012052.
[44] Lasheras FS, de Cos Juez FJ, Sánchez AS, Krzemień A, Fernández PR. Forecasting the COMEX copper spot price by means of neural networks and ARIMA models. Resources Policy. 2015;45:37–43.
[45] Manujakshi B, Kabadi MG, Naik N. A hybrid stock price prediction model based on pre and deep neural network. Data. 2022;7(5):51.
[46] Ahmed K, Ewees AA, Aziz MAE, Hassanien AE, Gaber T, Tsai PW, et al. A hybrid krill-ANFIS model for wind speed forecasting. In: International Conference on Advanced Intelligent Systems and Informatics. Springer; 2016. p. 365–72.
[47] Kristjanpoller W, Hernández E. Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. Expert Syst Appl. 2017;84:290–300.
[48] Fatima S, Uddin M. On the forecasting of multivariate financial time series using hybridization of DCC-GARCH model and multivariate ANNs. Neural Comput Appl. 2022;34(24):21911–25.
[49] Alzaeemi SA, Sathasivam S, Ali MKbM, Tay K, Velavan M, Hybridized intelligent neural network optimization model for forecasting prices of rubber in Malaysia. Comput Syst Sci Eng. 2023;47(2):1471–91.
[50] Chen ZY. A computational intelligence hybrid algorithm based on population evolutionary and neural network learning for the crude oil spot price prediction. Int J Comput Intell Syst. 2022;15(1):68.
[51] Sharma DK, Hota H, Brown K, Handa R. Integration of genetic algorithm with artificial neural network for stock market forecasting. Int J Syst Assurance Eng Manag. 2022;13(Suppl 2):828–41.
[52] Hadwan M, Al-Maqaleh BM, Al-Badani FN, Khan RU, Al-Hagery MA. A hybrid neural network and box-jenkins models for time series forecasting. CMC-Comput Mater Contin. 2022;70:4829–45.
[53] Sun K, Huang SH, Wong DSH, Jang SS. Design and application of a variable selection method for multilayer perceptron neural network with LASSO. IEEE Trans Neural Networks Learn Syst. 2016;28(6):1386–96.
[54] Al-Betar MA, Awadallah MA, Doush IA, Alomari OA, Abasi AK, Makhadmeh SN, et al. Boosting the training of neural networks through hybrid metaheuristics. Cluster Comput. 2023;26:1821–43.
[55] Tsantekidis A, Passalis N, Tefas A, Kanniainen J, Gabbouj M, Iosifidis A. Forecasting stock prices from the limit order book using convolutional neural networks. In: 2017 IEEE 19th Conference on Business Informatics (CBI). vol. 1. IEEE; 2017. p. 7–12.
[56] Mirjalili S. How effective is the Grey Wolf optimizer in training multi-layer perceptrons. Appl Intell. 2015;43(1):150–61.