Improving Stock Price Prediction through a Multilayer Perceptron Driven by a Grasshopper Optimization Algorithm: An Analysis of the Hang Seng Index

Lu Xia
Anhui Sanlian University, Hefei 230031, China
E-mail: xialuxialuxialu@126.com

Keywords: Hang Seng index, stock price, economic forecasting, multilayer perceptron, grasshopper optimization algorithm

Received: June 30, 2025

Given that time-series data is nonlinear, noisy, and dynamic, it may be difficult to predict stock prices in turbulent financial markets. To tune essential MLP hyperparameters such as the number of hidden units, learning rate, batch size, and epochs, this study presents a hybrid prediction model that combines a Multilayer Perceptron (MLP) with the Grasshopper Optimization Algorithm (GOA). On an 80/20 train–test split, the model is trained and tested on daily OHLC price and volume data from January 2015 to June 2023 for the Hang Seng Index (HSI). Benchmark models such as Transformer, Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Outlier-Robust Extreme Learning Machine (OR-ELM), Histogram-Based Gradient Boosting Regression (HGBR), and other hybrid optimizers (BBO-MLP, GA-MLP) are used for comparing performance. With an R^2 value of 0.9912, MAPE value of 0.83%, MAE value of 170.06, and MSE value of 48,618 on the test set, experimental results demonstrate that the suggested GOA–MLP achieved the highest accuracy according to all measures. MAPE and MAE were decreased by GOA–MLP by about 48.1% and 47.9%, respectively, in comparison with the best deep learning baseline (Bi-LSTM). With a total return of 96.52%, a maximum drawdown of 1.7%, and a Sharpe ratio of 3.14 far above that for a Buy & Hold strategy back testing the model in an artificial trading environment confirmed its effectiveness. The results show that the introduced GOA–MLP offers significant risk-adjusted performance enhancement under actual investment scenarios and improves predictive capability.

Povzetek: Študija predstavi hibrid GOA-MLP, kjer optimizator samodejno uglašuje hiperparametre MLP za napovedovanje cen iz šumnih, nelinearnih časovnih vrst ter s tem izboljša napovedno in trgovalno učinkovitost.

1 Introduction

Unpredictable changes in stock prices are because many factors that are interrelated to this behavior. Some of the possible causes influencing this phenomenon can be global economic indicators, changes in unemployment, monetary policy formulated by influential countries, immigration policy, natural calamities, public health situations, and so on. All participants in the stock market now aim to earn maximum income and reduce risk through thorough analysis. The collection of varied data, its integration into a logical framework, and the creation of models that can be trusted for precise forecasting, however, continue to be formidable obstacles. For businesses, investors, and equity traders looking to profit from upcoming market movements, stock price forecasting is a complex and difficult task. Timely and accurate forecasting is particularly challenging in the stock market due to its inherent characteristics, which include high noise levels, non-parametric patterns, nonlinear dependencies, and elements of deterministic chaos [1]. Even if the future were considered entirely uncertain and randomly unfolding day by day, it would yet fall into the realm of prediction, allowing one to foresee events and practically secure profit from them. The AI and ML methods to forecast stock market movements constitute one such approach. Even though the stock market is highly unpredictable, it is still good practice to feed in AI-generated forecasts as input before committing investment funds [2]. An Artificial Neural Network (ANN) is an algorithm specifically developed to comprehend complex problems that cannot be solved by simple ML algorithms or conventional neural networks. ANNs possess a higher level of complexity and sophistication in their relationships than the human brain. The method employs algebraic equations to guide data toward a model or time-series line. Undoubtedly, ANN is a highly popular ML technology with extensive application across disciplines. The technique has demonstrated extraordinary performance in interpreting the correlation between input and output data of complex physical processes. The growing applicability of hybrid predictive models, which fuse the merits of several approaches to improve forecasting accuracy, has been accentuated by recent developments in ML, particularly for complex, nonlinear domains like financial time series. A Grey Wolf Optimizer (GWO) method was proposed by Sneha S. et al. [3] to improve stock price prediction by automatically identifying the best parameters for GARCH and ARIMA models. Their method, compared to hand-tuned conventional models, improved efficiency in forecasting by 5% to 8%. Essam H. Houssein et al. [4] outperform many recent metaheuristics with their hybrid forecasting model, employing support vector regression alongside the equilibrium optimizer (EO-SVR) for predicting closing prices on the Egyptian Exchange. Their analysis demonstrates that little importance is placed on statistical values and technical indicators in prediction performance. For DJIA index stock price forecasting, Burak Gülmez suggests an optimized deep LSTM model using the Artificial Rabbits Optimization (ARO) algorithm (LSTM-ARO). From the outcome of other evaluation metrics, the LSTM-ARO model provides better performance than other neural networks and optimization models [5]. Supported by cutting-edge training methods and Principal Component Analysis (PCA)-based feature selection, Heng Lyu [6] suggests a hybrid ARIMA–LSTM stock market forecasting model that integrates linear and nonlinear pattern recognition. Ernest Kwame Ampomah et al. [7] use a variety of feature extraction and scaling techniques to test the performance of the Gaussian Naïve Bayes (GNB) algorithm in predicting stock price movement. GNB outperforms other GNB-based models on several important metrics in their findings and offers a considerable enhancement in predictive accuracy when combined with Linear Discriminant Analysis (LDA) and Min-Max scaling. Ramzi Saifan discusses, with the Quantopian simulator, the performance of three prediction and daily stock market trading ensemble machine learning techniques: Extremely Randomized Trees, Random Forest, and Gradient Boosting. The paper highlights the great potential of ensemble methods in automated trading systems by showing that all models with technical indicators trained produce substantial returns and high alpha values [8]. Ernest Kwame Ampomah uses NYSE, NASDAQ, and NSE data to study how well tree-based AdaBoost ensemble machine learning algorithms predict the stock market. AdaBoost-ExtraTrees (Ada-ET) outperformed all other algorithms on a range of evaluation measures in the experiment, demonstrating the power of ensemble techniques in predictive performance [9].

Using an ensemble of LSTM-based recurrent neural networks (RNNs) in exchange market forecasting, Algirdas Maknickas and Nijolė Maknickienė suggest a decision support system to investors [10]. The system uses distribution-based prediction methods like high-low, daily-weekly, and UK-NY timing to improve the signal recognition and set the limits. These methods were best for successful, short-term trading in volatile currency markets. The concept of biological neural networks in the brain is primarily inspired by the work of Rosenblatt in 1958. The Multilayer Perceptron (MLP) is a highly utilized form of ANN for constructing data-driven models [11]. Essentially, MLP is comprised of several artificial neurons or computational nodes [12]. MLP consists of three types of layers: the input layer, the hidden layer, and the output layer. Each layer of the neural network consists of linked neurons, which are coupled using weights and biases. Although MLPs have shown promise in a variety of fields, their effectiveness is largely dependent on the choice and initialization of hyperparameters, which can result in slow convergence or entrapment in local minima if done incorrectly.

To resolve design issues that are encountered in the actual world, metaheuristic algorithms that incorporate stochastic operators are increasingly being utilized in engineering [13], [14]. However, deterministic algorithms are not very successful at discovering global optimal solutions because they tend to become stuck in local optimal solutions [15]. Deterministic algorithms are considered to be reliable. Stochastic optimization algorithms, like evolutionary algorithms, can avoid local solutions and find global optimal solutions inside search spaces by using randomization as a core strategy [16]. When it comes to avoiding local optima, these approaches perform better than deterministic algorithms, even though each cycle of these approaches has the potential to give a different result. For instance, the implementation of genetic algorithms (GA) [17], ant lion optimization (ALO) [18], slime mold algorithms (SMA)[19], biogeography-based optimization (BBO)[20], and moth-flame optimization (MFO) [21] at this point. As stated before, the use of metaheuristic optimization algorithms for parameter optimization is explored in this paper to evade the setbacks of conventional MLP training approaches, which are frequently plagued by slow convergence and entrapment in local minima. To improve the learning ability of MLP, this research specifically examines three evolutionary optimizers: BBO, GA, and Grasshopper Optimization Algorithm (GOA). In unstable financial markets like the Hang Seng Index, these algorithms are used to tune the network weights and biases with the aim of accelerating convergence, avoiding poor local solutions, and enhancing prediction accuracy. The grasshopper is a perilous bug in the natural world that devastates plants and damages agricultural yields. According to Simpson et al. [22], grasshoppers are known to form one of the largest swarms in existence, even though they are typically observed as individual insects. The grasshopper undergoes a life cycle consisting of three distinct stages: egg, nymph, and adult. The phenomenon of food source swarming is an intriguing trait seen in grasshoppers, which served as the inspiration for the development of an optimization algorithm [23], [24]. In nature-inspired algorithms, the process of finding food sources often occurs in two stages: exploration and exploitation. In the case of the Grasshopper Optimization Algorithm (GOA), the exploitation process moves slowly, whereas the exploration process moves quickly [25]. The grasshopper naturally engages in both exploration and exploitation. [24]. Subsequently, this research modeled them to create the GOA model. The research aims to (i) improve prediction accuracy over benchmark models; (ii) improve robustness and convergence stability in volatile market conditions; and (iii) validate the model in a realistic investment setting through back testing, which includes cumulative return, drawdown, and Sharpe ratio analysis. Comparative tests against a variety of benchmarks demonstrate that the GOA–MLP meets and exceeds these objectives, offering significant improvements in both predictive accuracy and actual trading performance. Following this introduction, the document is formatted as follows: Section 2 of the investigation involves investigating the data source in detail and its associated aspects, which is just one of the analytical approaches used in the study. Section 3 elaborates on the obtained results and the pertinent discourse. The key conclusions are then succinctly outlined.

2 Methods and materials

2.1 The proposed model's outline

To predict the Hang Seng Index (HSI), this study suggests a hybrid approach that includes a MLP neural network and the GOA. The structure of the GOA-MLP model comprises two well-harmonized parts, as shown in Figure 1: the GOA tunes the network parameters for improved learning and convergence, and the MLP learns intricate patterns from the stocks. OHLC (Open, High, Low, Close) price and volume of the HSI from January 2015 to June 2023 were used to train and test the model; 80% of the data were used for training and 20% for testing. The predictive accuracy was greatly improved by using GOA as a metaheuristic optimizer. This allowed the model to avoid local minima and better adapt to the chaotic and nonlinear characteristics of financial time series. According to the results, the GOA-MLP hybrid model offers a reliable and effective framework for predicting the stock market in volatile conditions.

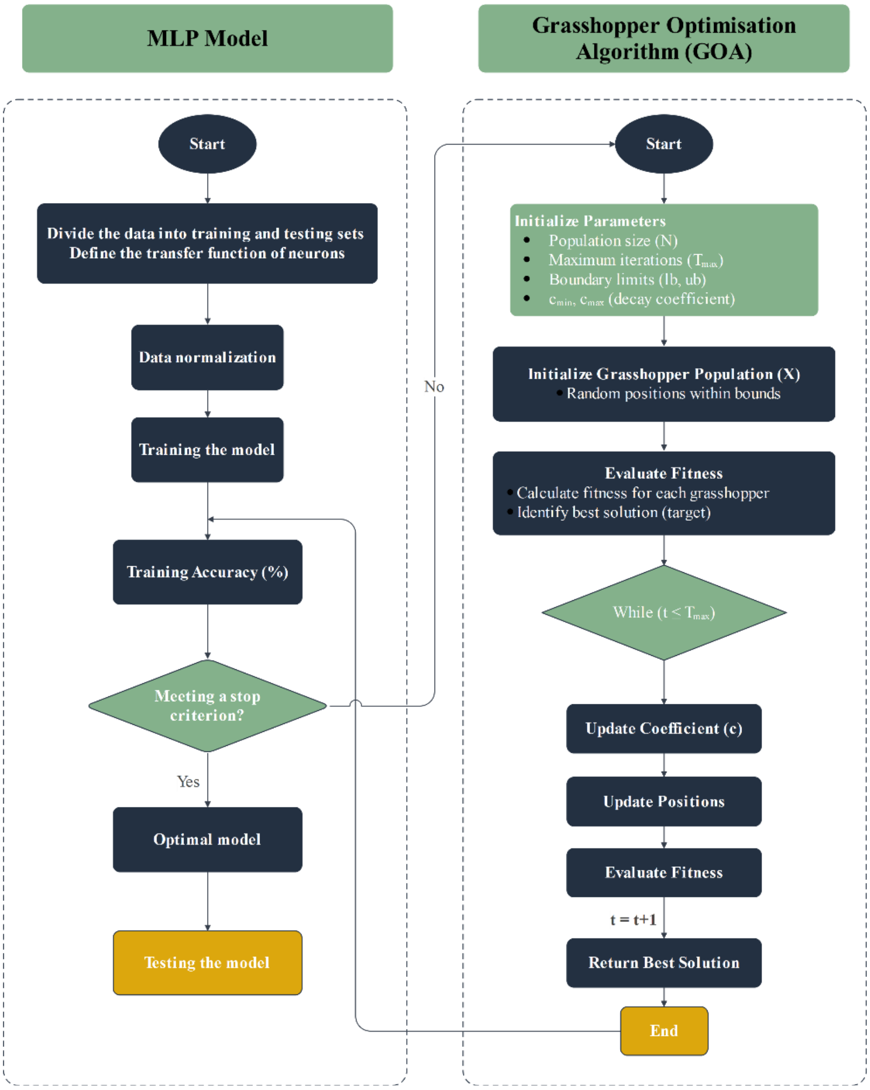


Figure 1: The GOA-MLP flowchart.

2.2 Biogeography based optimization

In 2008, Dan Simon introduced the BBO algorithm [20] as an innovative meta-heuristic method that involves transferring organisms between islands to find the best possible environment. The BBO technique uses the habitat suitability index (HSI) to evaluate the efficacy of a remedy. A solution with a high HSI is deemed favorable, whereas a solution with a low HSI is deemed unfavorable. The BBO algorithm has two fundamental operations: migration and mutation. The migration technique is utilized to improve poor solutions by transferring attributes from superior solutions (solutions of superior quality) onto the suboptimal ones. The mutation technique is utilized to stochastically modify one or more attributes of the solution, based on a predetermined probability. Mutation has the potential to enhance the variety within a population while also impeding the algorithm's progress towards a stable state. The following equations can be employed to get the immigration and emigration rates (I and k, respectively) for each iteration of the improvement loop:

$$\mu_k = \frac{E \times k}{n} \lambda_k = I\left(1 - \frac{k}{n}\right) \tag{1}$$

The rate at which individuals depart the habitat is represented by the symbol μ_k, which stands for the exit rate. The migration rate of the k^{th} habitat can be represented by the symbol λ_k. I: The highest possible migration rate can be attained. The condition $n = S_{max}$ is used to define the maximum number of species that a habitat is capable of supporting. To denote the highest possible migration rate, the letter E has been assigned. K: total number of species.

2.3 Genetic algorithm

J.H. Holland developed the genetic algorithm (GA), a metaheuristic algorithm, in 1992 [26], [27]. It is influenced by the evolution of biology. Makes use of initial point mutations and cross-overs to maximize a set of goals that are suggested for a thorough comprehension of these methods, which includes maximizing hyper-parameters. Furthermore, a GA accesses the objective function's results to determine which locations are the most interesting from an optimization perspective. The population is the basis of the GA algorithm. This suggests that a single population with a few chromosomes is where GA begins. For any possible combination of chromosomes, there is only one solution. GA can rate the performance of each chromosome by ranking each solution based on the fitness function [28].

Furthermore, a new population is generated by GA procedures (selection, crossover, mutation). GA creates and rates new populations by iteratively carrying out the previous procedure. The fitness function of each new solution determines its performance. Ultimately, GA will be ended under specific circumstances, and the best possible settlement will be found.

2.4 Grasshopper optimization algorithm

The swarm intelligence optimization method is very capable of effectively solving complex problems and has gained significant popularity in the last twenty years due to advancements in computer technology. To address this issue, the GOA is utilized. The GOA was introduced by Shahrzad Saremi in 2017 [24]. Upon its proposal, this new intelligent optimization technique quickly garnered significant interest. It has been applied in several domains due to its notable effectiveness and straightforward functionality [29], [30], [31], [32], [33]. The GOA is a nature-inspired algorithm, to according Figure 2, that was proposed based on the social interaction of grasshoppers. Figure 3 depicts this optimizer's entire process.

The subsequent model mathematically depicts the swarming behavior of grasshoppers:

$$X_i = r_1 G_i + r_2 A_i + r_3 S_i \tag{2}$$

Xi denotes the current position of the i-th grasshopper. The values $r_i (i = 1,2,3)$ are random numbers between 0 and 1, which are used to introduce randomness in their behavior.

The gravitational force experienced by the $\imath$th grasshopper at birth is denoted as G_i and may be computed using the following formula:

$$G_i = -g\hat{e_g}. \tag{3}$$

You may find the wind advection force on the $\imath$th grasshopper, A_i, by applying the subsequent equation:

$$A_i = u\hat{e_w} \tag{4}$$

The following formula may be used to calculate the social aptitude of the $\imath$th grasshopper, which is represented as S_i:

$$S_i = \sum_{j=1, j\neq i}^{N} s(x_j - x_i) \tag{5}$$

$\widehat{e_g}$ and $\widehat{e_w}$ denote unit vectors in distinct directions where g and u are constants. The distance between the j-th and i-th grasshoppers is shown by the symbol $|x_j - x_i|$, whereas N is the number of grasshoppers. The social interaction function is defined as the s in Eq. (5):

$$s(r) = fe^{-t} - e^{-r}, \tag{6}$$

Where f and l stand for the attraction's strength and endurance. The mathematical model of the GOA is presented in its ultimate form after undergoing further modification.

$$X_i^d = c\left(\sum_{j=1, j\neq i}^{N} c\frac{ub_d - lb_d}{2} s(x_j - x_i)\right) + T_d, \tag{7}$$

Where ub_d and lb_d denote the upper and lower bounds respectively, $\widehat{T_d}$ represents the current best solution, and c is a shift coefficient explained as:

$$c = cmax - l\frac{cmax - cmin}{L}. \quad (8)$$

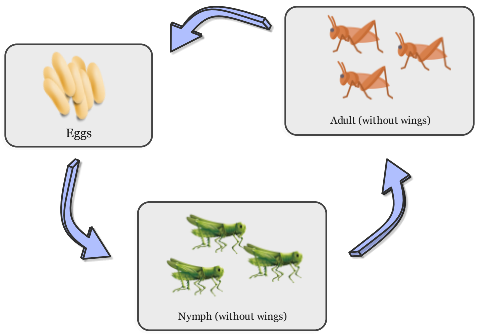


Figure 2: Swarm of grasshopper

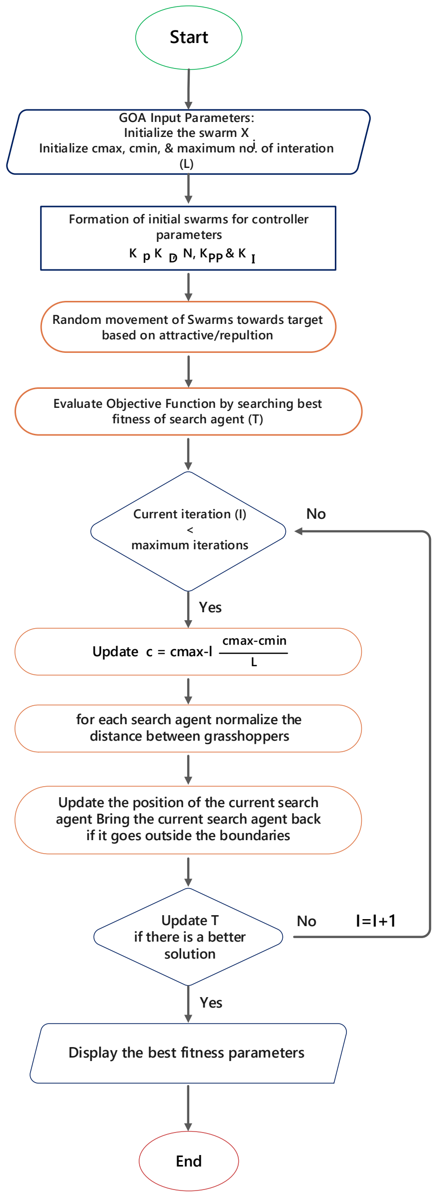


Figure 3: The flowchart of the GOA.

Every optimization algorithm employed in this study has its distinctive contribution to its ability to improve the learning capacity of the MLP. BBO provides an efficient mechanism for global search by harnessing the migration and mutation of fit solutions across habitats. GA provides an exploration-exploitation balance through artificial simulation of natural evolutionary processes via crossover, mutation, and selection. According to the swarming behavior of grasshoppers, GOA uses a well-defined social interaction model that demonstrates better dynamic adjustments in the search process. Compared with GA and BBO, GOA uses a nonlinear coefficient to gradually decline during the phase transition process from exploration to exploitation. This feature enables an algorithm to efficiently avoid premature convergence and explore the complicated solution space. Therefore, the parameters of the MLP can be adjusted more accurately and smoothly using GOA.Benchmark models

2.5.1. Outlier robust extreme learning machine

The OR-ELM is a variant of the Extreme Learning Machine that reduces outlier sensitivity with regularization or robust loss functions. This improves prediction performance and generalization in noisy data with the low complexity and fast training rate of ELM [34].

2.5.2. Transformer

The Transformer model allows for highly parallelizable training and robust performance on a vast variety of time-series forecasting tasks by learning long-range dependencies in sequential data without recurrence through the adoption of a self-attention mechanism [35].

2.5.3. Histogram-based gradient boosting regression

Histogram-Based Gradient Boosting Regression, or HGBR, is an ensemble method based on a decision tree that uses histogram-based binning of continuous features to speed up gradient boosting with the focus on high predictive accuracy [36].

2.5.4. Long short-term memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network that can learn long-term dependencies in sequence data using gating units and memory cells that prevent vanishing gradient problems [37].

2.5.5. Bidirectional long short-term memory

Bi-LSTM networks improve sequence modeling performance by reading input sequences in both the forward and backward directions, thereby encoding bidirectional temporal dependencies [38].

2.5.6. Multilayer perceptron

MLP models are recognized as a prominent category of ANN models. Their actions were guided by both human cognition and artificial intelligence [39]. The MLP model has individual computational units known as neurons. Every neuron is situated within distinct layers. The first layer is accountable for accepting inputs. The concealed levels get the input layers' income indicators [40], [41], [42]. The neuron's output is determined by activation functions. Each layer's neurons are closely connected to those of their adjoining layers. The last layer generates the outputs by functioning as the output layer. The MLP model uses the function that adds up all the results to calculate the total value obtained by multiplying every input:

$$Z_j = \sum_{i=1}^{n} \omega_{ij} I_i + \beta_j \tag{9}$$

n denotes the aggregate quantity of input neurons. I_i represents the input, β_j represents the bias, The weight of the link between the i_{th} node in the input layer and the j_{th} node in the hidden layer is indicated by the variable ω_{ij}. The variable Z_j represents the sum function. The outcomes of Eq. (9) are passed on to the activation function. Prior studies have shown that the sigmoid function is efficient in handling data within the MLP model, as evidenced by the studies conducted by Seifi et al. [43]and Jalali et al. [44]. The MLP model's structure is depicted in Figure 4.

Here is how the activation is used by the model:

$$f_j(x) = \frac{1}{1 + e^{-z}} \tag{10}$$

The function $f_j(x)$ represents the activation function. The ultimate result is calculated in the following manner:

$$\text{out}_i = \left(\sum_{i=1}^{n} \omega_{ij} I_j + \beta_J \right) \tag{11}$$

Which out_i is the last output of the MLP. These models generally incorporate conventional training processes, such as the gradient technique to determine the MLP's parameters, which include biases and weights. These algorithms may converge very slowly, or worse, get trapped in a local optimum.

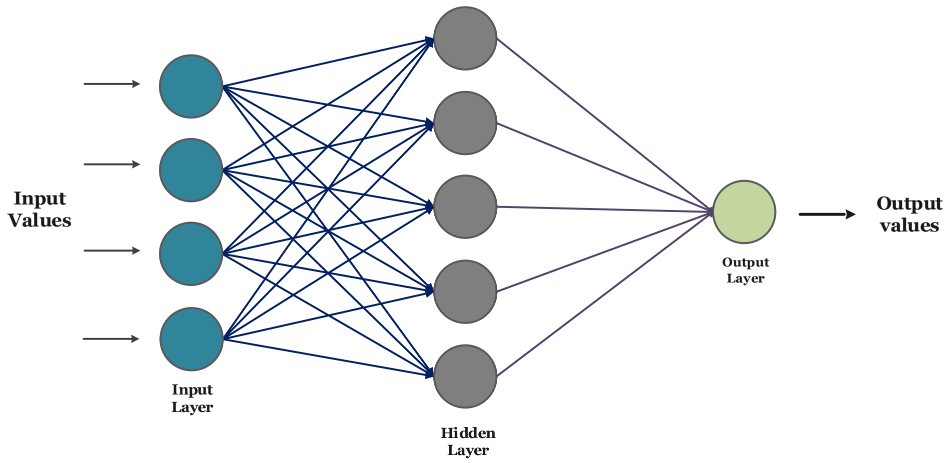


Figure 4: The methodology's structure of MLP.

The number of hidden units, training epochs, batch size, and learning rate are the most important hyperparameters tuned in this study. The search space and the final values chosen by each optimization algorithm (BBO, GA, and GOA) appear in Table 1. To find the setup with the minimum prediction error, each optimizer independently searched in the given parameter space. Its enhanced predictive accuracy is attributed to the GOA's choice of parameters, including fewer hidden units, moderate epochs, and a smaller learning rate. These parameters strike a balance between training stability and convergence rate. The GOA, which used a population size of 100 candidate solutions to search the hyperparameter space, chose the configurations that were used to train the MLP. To ensure steady convergence and fair comparison between experiments, the number of training epochs of the optimizer was fixed at 500. Selected configurations were used to train the MLP for 450 epochs. These were the number of hidden units, the learning rate, the batch size, and the number of epochs used for training. To make sure that the same results were generated each time, a fixed random seed (seed = 42) was used for all experiments, including initialization of weights, data shuffling, and optimization.

Table 1: Hyperparameter search space and optimal values for optimizing the MLP model by the BBO, GA, and GOA optimization algorithms.

MLP		BBO	GA	GOA
n_hidden_units	[2, 64]	8	16	8
epoch	[100, 1000]	600	650	450
batch_size	[2, 32]	16	8	8
learning_rate	[0.0001, 1]	0.1	0.05	0.01

2.5 Data collection and preparing

Incorporating the number of volumes, as well as the Open, High, Low, and Close prices, within a specific time frame is crucial for conducting a thorough analysis. The data for this investigation were collected from the Hang Seng Index (HSI), ranging from January 2, 2015, to June 29, 2023. The HSI is a famous market index that tracks a selection of major companies featured on this Exchange. The HSI was chosen as the case study because of its substantial worldwide influence and high volatility. The HSI is a prominent market index in Asia that tracks the performance of large Hong Kong-listed companies, many of which are active in global markets. A complex and dynamic benchmark for stock market prediction, the index is extremely sensitive to regional policy changes, international economic trends, and geopolitical tensions. These qualities offer a demanding testing environment for assessing the resilience and versatility of forecasting models such as the suggested GOA-MLP. The HSI comprises a wide variety of prominent Hong Kong companies. It includes finance, real estate, technology, telecommunications, manufacturing, etc. Market capitalization determines the weight of each company's stock in the index. The HSI of the Hong Kong stock market closely tracks the domestic economy. This index is widely used to assess Hong Kong's financial health and investor outlook. Index fluctuations might affect investor perceptions of regional economic conditions. The index includes many multinational companies with a worldwide reach. The index's global reach makes it a key economic and market indicator beyond Hong Kong. To represent the changing Hong Kong market, the HSI constituents are continuously examined and adjusted. The index may reject companies that don't meet its standards and include new ones that do. The HSI may be affected by market laws, economic conditions, government policies, and global events. In short, the HSI tracks the major companies listed on the HSI Exchanges. Economic data is insightful to the Hong Kong Economic Index, sets standards for investors, and helps to understand the mood and trend of the stock market. OHLC price and volume data were supplied to the model as training data. One of the key aspects to achieving the success of ML models is proper data preparation. Data cleaning is the first step in the process, where errors, missing values, and inconsistencies are identified and corrected in the dataset. It is a crucial process because poor-quality data can affect model performance and result in incorrect predictions. The overall accuracy and resilience of the model are greatly improved by ascertaining and fine-tuning the dataset exhaustively before training. Designing leading ML models starts with a clean, stable dataset. Normalization is then used after cleaning to normalize the input features into a similar order of magnitude, usually between 0 and 1. By causing all features to make a similar contribution to the learning process, this normalization stops models from favoring variables whose scales have larger values. Additionally, normalization speeds up model convergence and improves its ability to identify inherent patterns. Normalization is done through the following equation [45]:

$$Xscaled = \frac{(X - Xmin)}{(Xmax - Xmin)} \tag{12}$$

Finally, data splitting is used for testing the model's generalization ability. Data is split into two sets: 20% is left for testing, and the remaining 80% is utilized to train the model. This portion is used to make sure the model acquires generalizable patterns and does not memorize the training set by preventing overfitting and assisting in the measurement of the model's performance against new data.

3 Results and discussions

3.1 Assessment metrics

The accuracy of the projections was appraised using several performance criteria.

The R^2 scores are quantitative measures assessing to what extent a model accurately representing the data. When the score tends to become 1, the quality of the model increases; hence this measure is used to analyze the accuracy of the model [46].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{13}$$

Difference of data observed with data expected is calculated with the MSE (Mean Squared Error). To get the numerical value, one must first identify the square of the difference between the observed and expected values. Then, estimate the average of all squared differences. The figure given displays the accuracy of the model, with a lower MSE indicating higher accuracy [47].

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{k}=0}^{\mathrm{n}} \binom{\mathrm{n}}{\mathrm{k}} (\mathrm{Fi} - \mathrm{Yi})\mathrm{b}^2 \tag{14}$$

The Mean Absolute Error (MAE) is a complementary statistic that we may consider to measure how far an observation is from the prediction. The performance of the model depends upon this statistic; the MAE should go down for better accuracy [47].

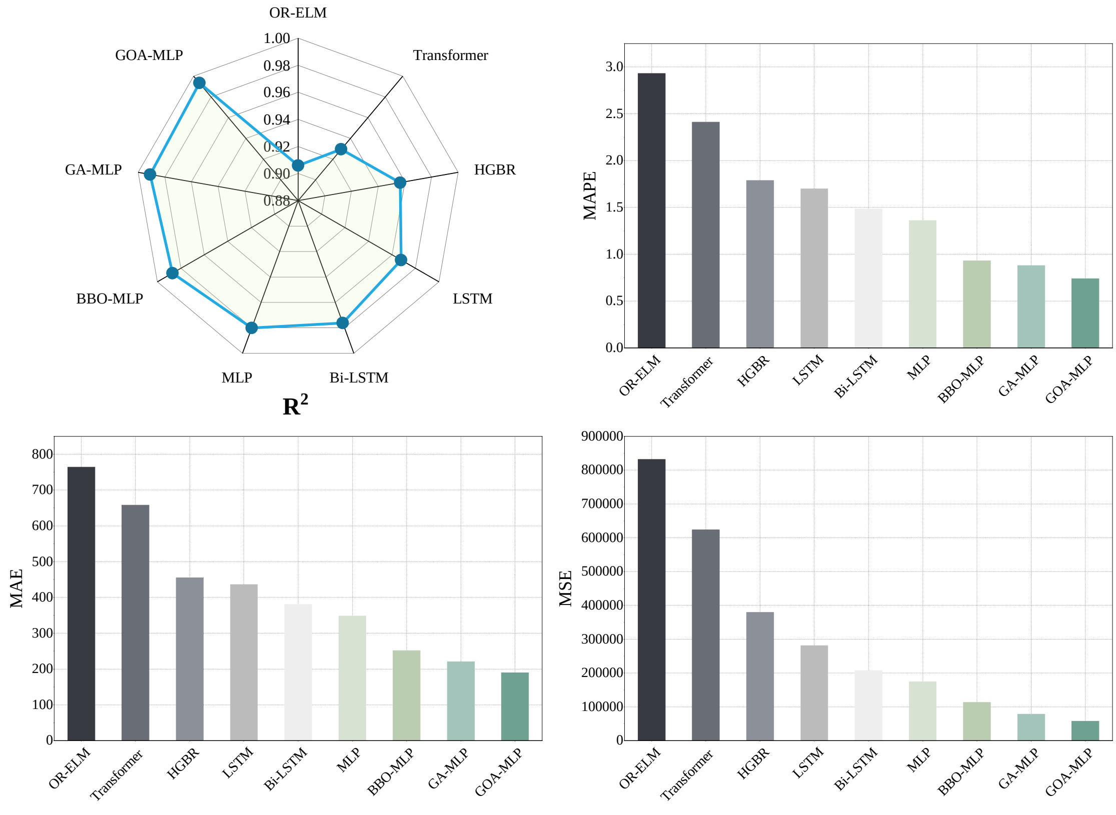
$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{15}$$

A method of checking a model for precision would be to use MAPE, or Mean Absolute Percentage Error. To determine the percentage difference between predicted and actual values, first an absolute difference is determined and then divided by the value of observation. These percentages are then subjected to an averaging operation. It must be clear that this number aims at analyzing the accuracy of the model. Less MAPE will mean more precision [46].

$$MAPE = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) \times 100 \tag{16}$$

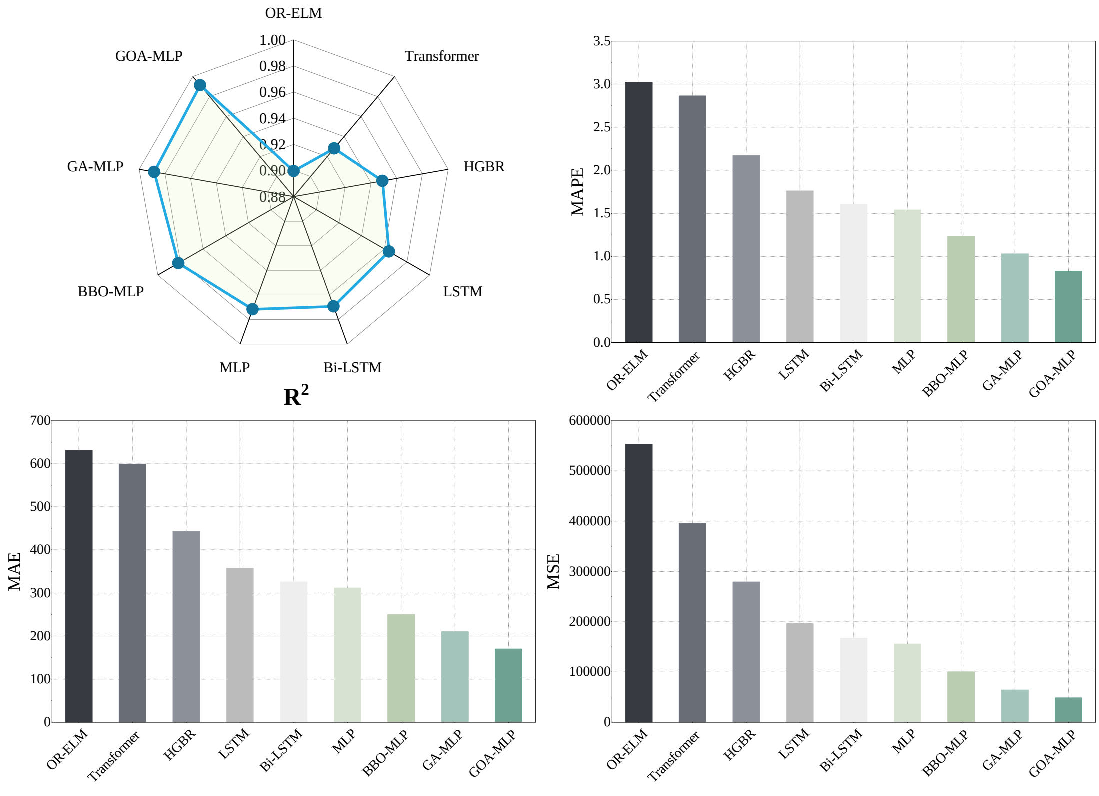
3.2 Analysis and comparisons

The primary objective of the investigation is to create and assess the best hybrid algorithm available for stock price prediction. Predictive models have been developed as an outcome of extensive research on the intricate aspects influencing stock market movements. This was carried out in order to give analysts and investors trustworthy information so that they could make well-informed financial decisions. A comprehensive analysis of the performance of each method is presented in Table 2, and Figs. 5 and 6, respectively.



(a)

Figure 5: Values for the assessment metrics during training set for the models.



(b)

Figure 6: Values for the assessment metrics during testing set for the models.

A comprehensive comparison of various prediction models, sophisticated deep learning architectures, and traditional machine learning models is discussed in Table 2. Four evaluation metrics R^2, MAPE, MAE, and MSE are used to analyze the model's performance over training and testing datasets. A comprehensive assessment framework of this nature ensures that the analysis includes the error magnitude, generalization ability, and point prediction accuracy. The relatively low R^2 values (≤ 0.9565) and higher error measures of the OR-ELM and HGBR models indicate relatively poor predictive performance. Although computationally efficient, these methods' use of shallow learning architectures restricts their ability to model the complex nonlinear dependencies, chaotic volatility, and long-range temporal dependencies of financial time series. While the Transformer model is theoretically powerful in modeling sequential dependencies using self-attention mechanisms, it only achieves modest improvement over conventional approaches (R^2 = 0.9282, test MAPE = 2.86). This is probably because it is susceptible to data paucity and over-parameterization in noisy, nonstationary settings like equity markets. By efficient vanishing gradient reduction and making use of temporal context, the LSTM and Bi-LSTM architectures that are capable of learning long-term dependencies surpass the common baselines. By using both past and future contextual information within sequence modeling, the bidirectional version, Bi-LSTM, provides a higher predictive accuracy (test R^2 = 0.9694, MAPE = 1.60) than its unidirectional LSTM. However, the recurrent models have higher error rates than the optimization-enhanced MLP variants, indicating suboptimal convergence and possible overfitting to temporal noise even with enhanced sequential modeling. The baseline MLP also learns competitively with nonlinear transformations in deep hidden layers (test R^2 = 0.9717, MAPE = 1.54), but still retains its architecture and learning parameters fixed by conventional tuning that remains vulnerable to shallow local minima. Metaheuristic optimization significantly improves prediction performance: both BBO-MLP (test R^2 = 0.9817, MAPE = 1.23) and GA-MLP (test R^2 = 0.9883, MAPE = 1.03) have dramatic error reductions against the baseline MLP. These improvements reflect the contribution of global search heuristics to overcoming the multimodal, high-dimensional hyperparameter space that comes with optimizing neural networks. With test R^2 = 0.9912, MAPE = 0.83, MAE = 170.06, and MSE = 48,618, the GOA-MLP hybrid performs best on all four metrics. GOA-MLP significantly improves predictive precision and robustness by cutting test MAPE by about 48.1% and MAE by 47.9% against the best deep learning reference (Bi-LSTM). GOA-MLP demonstrates its optimization advantage by achieving additional reductions in MAPE (19.4%) and MAE (19.1%) even when compared with the performing competitor hybrid (GA-MLP).

Table 2: The outcomes of the methodologies

MODEL/Metrics	TRAIN SET				TEST SET			
	R^2	MAPE	MAE	MSE	R^2	MAPE	MAE	MSE
OR-ELM	0.9059	2.93	763.74	831725	0.8996	3.02	631.14	553422
Transformer	0.9294	2.41	658.17	623476	0.9282	2.86	598.90	395837
HGBR	0.9565	1.79	454.76	379568	0.9489	2.17	443.11	279132
LSTM	0.9678	1.70	435.90	281124	0.9641	1.76	357.76	196239
Bi-LSTM	0.9762	1.48	380.36	207730	0.9694	1.60	326.02	167418
MLP	0.9801	1.36	348.22	174192	0.9717	1.54	311.72	155726
BBO-MLP	0.9872	0.93	251.74	112962	0.9817	1.23	250.39	100199
GA-MLP	0.991	0.88	220.83	78593	0.9883	1.03	210.27	64366
GOA-MLP	0.9934	0.74	189.77	57231	0.9912	0.83	170.06	48618

Several algorithmic features in combination give rise to GOA-MLP's enhanced performance:

During the optimization time frame, GOA utilizes an adaptive nonlinear decreasing coefficient to control the transition from exploration (global searching) to exploitation (local refinement). This enables more vigorous exploration of the hyperparameter space and avoids premature convergence. The intragroup weight convergence dynamics of the MLP are directly influenced by the upper-level hyperparameters (number of hidden units, learning rate, batch size, and epochs) tuned for by GOA. This leads to improved generalization and less overfitting in noisy environments.

GOA uses the position-update approach with inspiration drawn from the behavior of grasshopper swarms that adaptively modifies candidate solutions based on fluctuations in the fitness landscape. This enables diversity maintenance while focusing on optimal regions. With fewer hidden units, fewer epochs of training, limited batch size, and a moderate learning rate, the final GOA-selected configuration reduces variance in parameter updates to produce smoother convergence curves and better performance on novel test data.

This exhaustive benchmark test shows GOA-MLP to far outperform competing metaheuristic-optimized neural networks as well as deep recurrent and standard recurrent frameworks. The ability of the model to navigate the volatility and structural complexity of financial markets with greater predictive accuracy arises due to the synergistic combination of swarm-intelligence-driven global search with MLPs' representational strength. Designing and validating a hybrid GOA–MLP model that can provide quantifiable gains over state-of-the-art (SOTA) stock price forecasting techniques was the main objective of this study. In comparison to the top-performing benchmark models, this study specifically sought to reduce MAPE and MAE by at least 15%. The GOA–MLP model surpassed this benchmark, as shown by the results and detailed in Table 2. It reduced MAPE and MAE by 19.4% and 19.1%, respectively, over GA–MLP and by approximately 48.1% and 47.9% when compared to the top deep learning baseline (Bi-LSTM) while maintaining a high R^2 of 0.9912 on the test set. Another goal was to improve resilience against volatile market conditions by leveraging GOA's adaptive exploration–exploitation mechanism to find hyperparameter settings that inhibit overfitting and induce more rapid convergence. Figure 7 and 8 demonstrate how the suggested approach may accurately forecast the stock market.

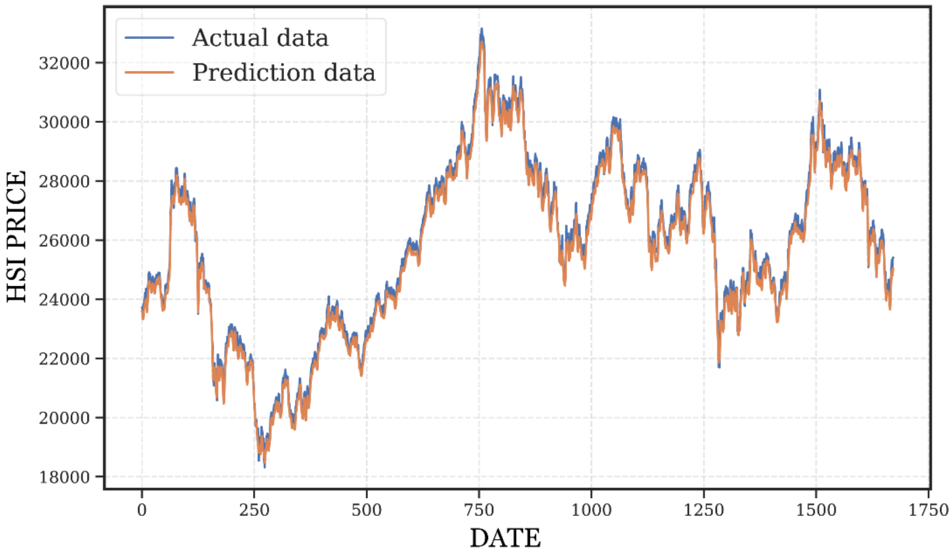


Figure 7: The GOA-MLP train data forecast curve.

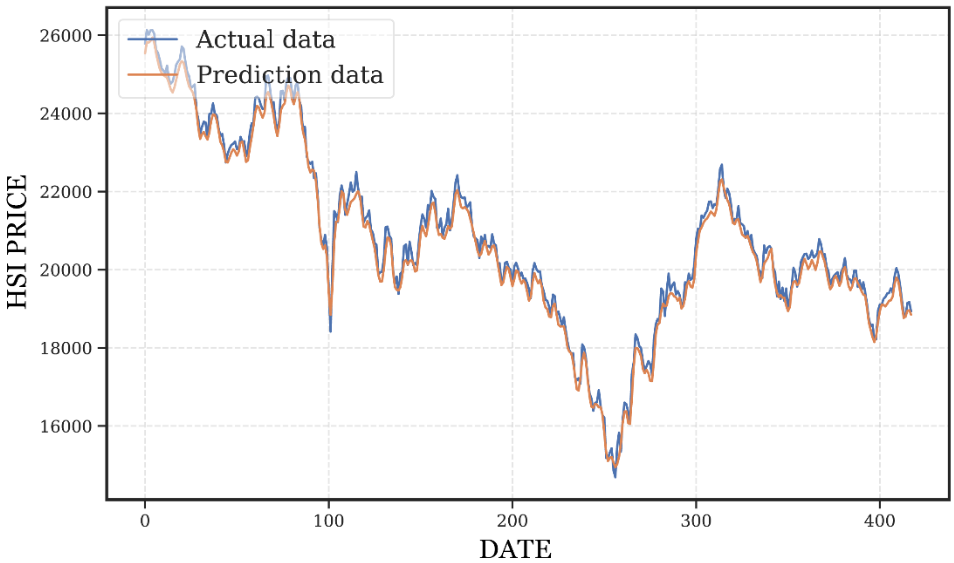


Figure 8: The GOA-MLP test data forecast curve.

3.3 Real-world back testing and risk analysis

A real-world back testing experiment was carried out to evaluate the suggested GOA–MLP model's practical usefulness beyond standard predictive accuracy metrics. To manage a simulated portfolio during the test period, the model's daily predictions were translated into trading signals, and the results were compared to a conventional buy and hold strategy on the same asset. The Sharpe ratio, maximum drawdown (%), and cumulative return (%) were the three-evaluation metrics taken into account. Together, these metrics evaluate risk-adjusted returns, profitability, and downside risk, providing a more thorough assessment of investment performance. According to Table 3, the trading strategy based on GOA and MLP produced a

Sharpe ratio of 3.14, a maximum drawdown of only 1.7%, and a cumulative return of 96.52%. Profitability and effective risk management are demonstrated by the high return and low drawdown. Conversely, the buy and hold strategy had a Sharpe ratio of -0.55, a maximum drawdown of 43.81%, and a negative cumulative return of -27.55% percent, all of which indicated unfavorable risk-return characteristics. These variations are further highlighted by the portfolio value trajectories displayed in Figure 9.

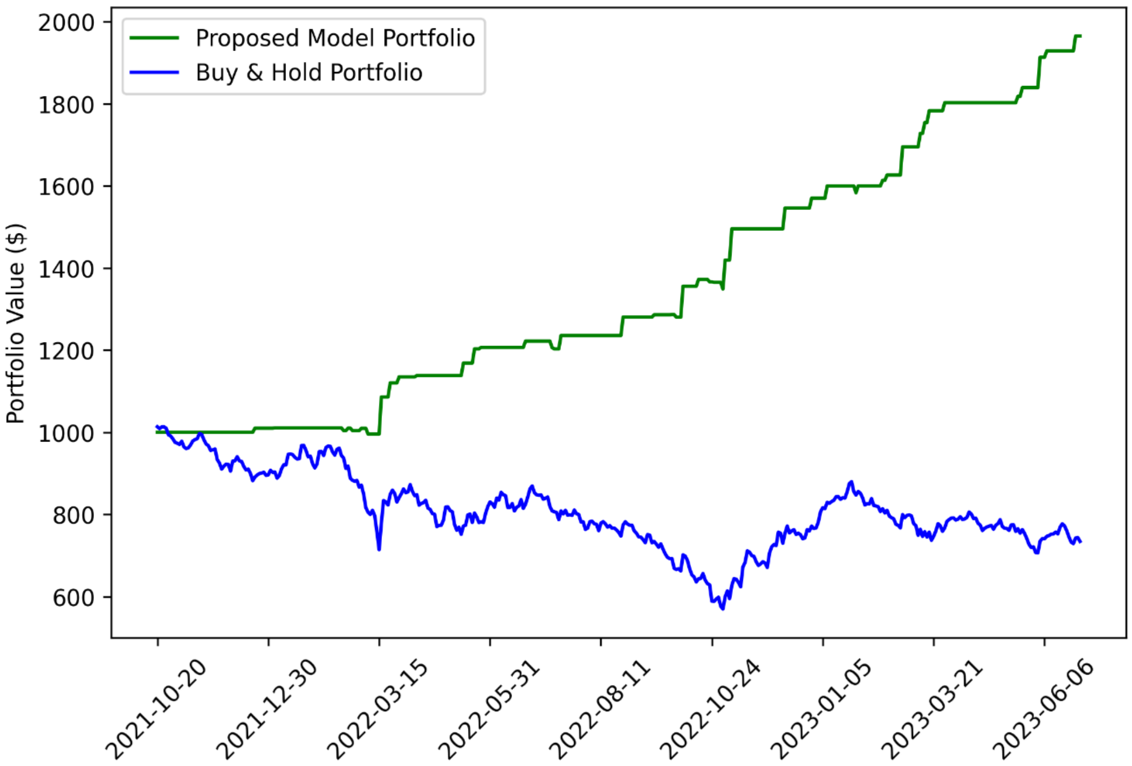


Figure 9. Trajectories of the portfolio values for the buy-and-hold strategy and the suggested GOA–MLP-based trading strategy during the test period

Throughout the back test period, the GOA–MLP strategy avoided significant capital drawdowns and continued to grow steadily, whereas the buy and hold strategy saw protracted declines and was unable to return to its starting point. These results show that in addition to achieving greater statistical predictive accuracy, the proposed GOA–MLP model also translates these gains into tangible investment benefits, such as appreciable improvements in risk-adjusted returns and capital preservation in volatile market conditions.

Table 3: Results of a back test comparing the suggested GOA–MLP-based trading strategy with a traditional buy and hold strategy during the test period

Strategy	**Cumulative Return (%)**	**Max Drawdown (%)**	**Sharpe Ratio**
Model-Based	96.52	1.7	3.14
Buy & Hold	-27.55	43.81	-0.55

4 Conclusion

For very long, prediction of stock prices has been a favorite expanse of investigation. Investors have been building ever-so-more accurate forecasting models to gather big bucks. Predicting stock market movements becomes really strenuous when there are multiple sources of uncertainty like legislative enactments and sociocultural situations including pandemics. For making correct forecasts, it would be essential to come to terms with the stochastic and non-linear aspects of the market. Fortunately, the GOA-MLP stands as a high accurate model and thus can aptly address the presented problems. Inclusively presented in this paper are the MLP, BBO-MLP, and GA-MLP models for stock price forecasting. OHLC price and volume data from the HSI shares were part of the data utilized for the research project. The dataset is from the start of 2015 to the end of 2023, covering a particular time interval. The analysis, while forecasting stock prices, reveals that the GOA-MLP model is highly competent in forecasting, with both excellent performance and consistent results.

- During the research, a comparative analysis was conducted to evaluate the GOA-MLP model in terms of its accuracy and ability to make accurate predictions compared to other models. The findings of this study provide evidence that the GOA-MLP model consistently outperforms other models. According to the test results, the average score of MSE shows that the level of accuracy in predictions is good. The average MAPE score of the model was 0.83, which shows that it had significant accuracy in its predictions during the study. Both the high R^2 value of 0.9912 and the low MAE score of 170.06 can be taken as evidence that the predictions are based on an accurate and consistent basis. Compared to other investigated models, the GOA-MLP model has shown higher performance in terms of accuracy and efficiency.

As mentioned earlier, GOA-MLP shows better performance compared to other models in competitive conditions. Further validation of the effectiveness of the suggested model in accurately and comprehensively forecasting volatility in the stock market is provided by the findings. By utilizing this approach, risk minimization is made easier, and investors are provided with the opportunity to make informed investment selections by analyzing the many data points.

Acknowledgments

This work is supported by the key project of Teaching Quality and Teaching Reform in Higher Education Institutions of Anhui Provincial Department of Education: Provincial first-class undergraduate construction site in accounting (No. 2020ylzy002), and accounting professional transformation and enhancement project (No. 2022zygzts036).

Ethical approval

The research paper has received ethical approval from the institutional review board, ensuring the protection of participants' rights and compliance with the relevant ethical guidelines.

References

[1] R. G. Ahangar, M. Yahyazadehfar, and H. Pournaghshband, "The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in Tehran stock exchange," *arXiv preprint arXiv:1003.1457*, Cornell University, 2010. https://doi.org/10.48550/arXiv.1003.1457.

[2] P. Chhajer, M. Shah, and A. Kshirsagar, "The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction," *Decision Analytics Journal*, Elsevier, vol. 2, p. 100015, 2022. https://doi.org/10.1016/j.dajour.2021.100015.

[3] N. Buduma and N. Locascio, "Deep Learning," 2017.

[4] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Appl Soft Comput*, Elsevier, vol. 11, no. 6, pp. 4135–4151, 2011. https://doi.org/10.1016/j.asoc.2011.02.032.

[5] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf Sci (N Y)*, Elsevier, vol. 237, pp. 82–117, 2013. https://doi.org/10.1016/j.ins.2013.02.041.

[6] A. R. Simpson, G. C. Dandy, and L. J. Murphy, "Genetic algorithms compared to other techniques for pipe optimization," *J Water Resour Plan Manag*, ASCE Library, vol. 120, no. 4, pp. 423–443, 1994. https://doi.org/10.1061/(ASCE)0733-9496(1994)120:4(423).

[7] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[8] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[9] S. Mirjalili, "The ant lion optimizer," *Advances in engineering software*, Elsevier, vol. 83, pp. 80–98, 2015. https://doi.org/10.1016/j.advengsoft.2015.01.010.

[10] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, Elsevier, vol. 111, pp. 300–323, 2020. https://doi.org/10.1016/j.future.2020.03.055.

[11] D. Simon, "Biogeography-based optimization," *IEEE transactions on evolutionary computation*, IEEE, vol. 12, no. 6, pp. 702–713, 2008. https://doi.org/10.1109/TEVC.2008.919004.

[12] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl Based Syst*, Elsevier, vol. 89, pp. 228–249, 2015. https://doi.org/10.1016/j.knosys.2015.07.006.

[13] S. J. Simpson, A. R. McCaffery, and B. F. Hägele, "A behavioural analysis of phase change in the desert locust," *Biological reviews*, Cambridge, vol. 74, no. 4, pp. 461–480, 1999. https://doi.org/10.1017/S000632319900540X.

[14] S. M. Rogers, T. Matheson, E. Despland, T. Dodgson, M. Burrows, and S. J. Simpson, "Mechanosensory-induced behavioural gregarization in the desert locust Schistocerca gregaria," *Journal of Experimental Biology*, Journal of Experimental Biology, vol. 206, no. 22, pp. 3991–4002, 2003. https://doi.org/10.1242/jeb.00648.

[15] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in engineering software*, Elsevier, vol. 105, pp. 30–47, 2017. https://doi.org/10.1016/j.advengsoft.2017.01.004.

[16] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evol Comput*, MIT Press Direct, vol. 4,

no. 1, pp. 1–32, 1996. https://doi.org/10.1162/evco.1996.4.1.1.

[17] J. H. Holland, “Genetic Algorithms Computer programs that" evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand,” *Sci Am*, pp. 66–72, 1992.

[18] J. Luo, H. Chen, Y. Xu, H. Huang, and X. Zhao, “An improved grasshopper optimization algorithm with application to financial stress prediction,” *Appl Math Model*, Elsevier, vol. 64, pp. 654–668, 2018. https://doi.org/10.1016/j.apm.2018.07.044.

[19] X. Xiang, X. Ma, M. Ma, W. Wu, and L. Yu, “Research and application of novel Euler polynomial-driven grey model for short-term PM10 forecasting,” *Grey Systems: Theory and Application*, Emerald, vol. 11, no. 3, pp. 498–517, 2021. https://doi.org/10.1108/GS-02-2020-0023.

[20] S. Łukasik, P. A. Kowalski, M. Charytanowicz, and P. Kulczycki, “Data clustering with grasshopper optimization algorithm,” in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Prague, Czech Republic, IEEE, 2017, pp. 71–74. https://doi.org/10.15439/2017F340.

[21] A. Fathy, “Recent meta-heuristic grasshopper optimization algorithm for optimal reconfiguration of partially shaded PV array,” *Solar Energy*, Elsevier, vol. 171, pp. 638–651, 2018. https://doi.org/10.1016/j.solener.2018.07.014.

[22] M. Ahanch, M. S. Asasi, and M. S. Amiri, “A Grasshopper Optimization Algorithm to solve optimal distribution system reconfiguration and distributed generation placement problem,” in *2017 IEEE 4th international conference on knowledge-based engineering and innovation (KBEI)*, Tehran, Iran, IEEE, 2017, pp. 659–666. https://doi.org/10.1109/KBEI.2017.8324880.

[23] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, “Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems,” *Applied Intelligence*, Springer, vol. 51, pp. 1531–1551, 2021. https://doi.org/10.1007/s10489-020-01893-z.

[24] M. Ehteram *et al.*, “Design of a hybrid ANN multi-objective whale algorithm for suspended sediment load prediction,” *Environmental Science and Pollution Research*, Springer, vol. 28, pp. 1596–1611, 2021. https://doi.org/10.1007/s11356-020-10421-y.

[25] F. B. Banadkooki, M. Ehteram, F. Panahi, S. S. Sammen, F. B. Othman, and E.-S. Ahmed, “Estimation of total dissolved solids (TDS) using new hybrid machine learning models,” *J Hydrol (Amst)*, Elsevier, vol. 587, p. 124989, 2020. https://doi.org/10.1016/j.jhydrol.2020.124989.

[26] F. B. Banadkooki *et al.*, “Enhancement of groundwater-level prediction using an integrated machine learning model optimized by whale algorithm,” *Natural resources research*, Springer, vol. 29, pp. 3233–3252, 2020. https://doi.org/10.1007/s11053-020-09634-2.

[27] A. Seifi, M. Ehteram, V. P. Singh, and A. Mosavi, “Modeling and uncertainty analysis of groundwater level using six evolutionary optimization algorithms hybridized with ANFIS, SVM, and ANN,” *Sustainability*, MDPI, vol. 12, no. 10, p. 4023, 2020. https://doi.org/10.3390/su12104023.

[28] S. M. J. Jalali, R. Hedjam, A. Khosravi, A. A. Heidari, S. Mirjalili, and S. Nahavandi, “Autonomous robot navigation using moth-flame-based neuroevolution,” *Evolutionary Machine Learning Techniques: Algorithms and Applications*, Springer, pp. 67–83, 2020. https://doi.org/10.1007/978-981-32-9990-0_5.