International Journal of Cognitive Computing in Engineering 4 (2023) 109–117

Contents lists available at ScienceDirect

International Journal of Cognitive Computing in Engineering

journal homepage: https://www.keaipublishing.com/en/journals/international-journal-of-cognitive-computing-in-engineering/

Stock price predictive analysis: An application of hybrid Barnacles Mating Optimizer with Artificial Neural Network

Zuriani Mustaffa[a,*], Mohd Herwan Sulaiman[b]

[a] Faculty of Computing, Universiti Malaysia Pahang, Pekan, Pahang 26600, Malaysia
[b] Faculty of Electrical & Electronics Engineering Technology, Universiti Malaysia Pahang, Pekan, Pahang 26600, Malaysia

ARTICLE INFO

Keywords:
Artificial Neural Network
Barnacles Mating Optimizer
Machine learning
Optimization
Stock price prediction
Time series prediction

ABSTRACT

Artificial Neural Network (ANN) is an effective machine learning technique for addressing regression tasks. Nonetheless, the performance of ANN is highly dependent on the values of its parameters, specifically the weight and bias. To improve its predictive generalization, it is crucial to optimize these parameters. In this study, the Barnacles Mating Optimizer (BMO) is employed as an optimization tool to automatically optimize these parameters. As a relatively new optimization algorithm, it has been shown to be effective in addressing various optimization problems. The proposed hybrid predictive model of BMO-ANN is tested on time series data of stock price using six selected inputs to predict the next day' closing prices. Evaluated based on Mean Square Error (MSE) and Root Mean Square Error (RMSPE), the proposed BMO-ANN exhibits significant superiority over the other identified hybrid algorithms. Additionally, the difference in means between BMO-ANN and other identified hybrid algorithms was found to be statistically significant, with a significance level of 0.05%.

1. Introduction

For decades, machine learning techniques have demonstrated their superiority in solving regression problems and producing better outputs than statistical techniques. This subfield of computer science involves training models on historical data for use in prediction tasks (Wang et al., 2020). Notable examples of machine learning techniques include Artificial Neural Networks (ANN), which imitate the human brain by using computational elements that simulate neurons and their connections (Portillo Juan & Negro Valdecantos, 2022). Another example is Support Vector Machines (SVM), which were developed based on statistical learning theory (Vapnik, 1995). Its variant, Least Squares Support Vector Machines (LSSVM), requires fewer optimization parameters compared to the original version (Suykens et al., 2002). The applications of machine learning techniques are widely spread across fields, including medical (Lalmuanawma et al., 2020; Shahid et al., 2020; Agrawal et al., 2022; Solayman et al., 2023), engineering (Yan et al., 2022; Shen et al., 2022), finance (Kara et al., 2011), smart homes (Saxena & Varshney, 2021), cybersecurity (Zhihua Cui et al., 2018), and many more. Machine learning techniques are capable of producing promising results due to their ability to capture the non-linear features present in complex real-world datasets. These datasets are influenced by various factors, resulting in non-linear features that require appropriate techniques for prediction. For instance, datasets such as crude oil prices are highly dependent on economic crises and political events (Öztunç Kaymak & Kaymak, 2022), while stock prices exhibit non-linearity and high volatility (Liu et al., 2022). Additionally, Covid19 outbreak cases are influenced by various uncertainties (Öztunç Kaymak & Kaymak, 2022), among others. Statistical techniques, such as Autoregressive Integrated Moving Average (ARIMA) (Swaraj et al., 2021) and Exponential Smoothing (ES) (Lai et al., 2006), are not suitable for these datasets due to their limitations in capturing the non-linear features. Therefore, the application of machine learning techniques offers a better approach for addressing this issue.

However, to obtain an accurate prediction result, it is essential to optimize the parameters of the machine learning technique for good generalization. Therefore, hybridization with optimization algorithms becomes a favourable option. Numerous researchers have presented existing works on hybrid models of optimization techniques with machine learning, including in addressing regression tasks. For instance, in Naderi et al. (2019), a crude oil price prediction model was proposed using the meta-heuristic Bat Algorithm (BA) to optimize the parameters of Least Squares Support Vector Machines (LSSVM), Genetic Programming (GP), ARIMA and ANN. Error analysis demonstrated the superiority of the hybrid model over the single methods. In a previous work by Karasu and Altan (2022), a hybrid model of LSTM with Chaotic Henry Gas Solubility Optimization was proposed for the same data field as in Naderi et al. (2019). On the other hand, a wind speed predictive model

* Corresponding author.
E-mail address: zuriani@ump.edu.my (Z. Mustaffa).

https://doi.org/10.1016/j.ijcce.2023.03.003
Received 7 December 2022; Received in revised form 3 March 2023; Accepted 3 March 2023
Available online 10 March 2023

was presented in Tian (2020), which is based on hybrid LSSVM model with an improved Firefly Algorithm (FA). A comparison against other prediction models showed the superiority of the proposed model.

In Yeh et al. (2023), a hybrid Convolutional Neural Network with Simplified Swarm Optimization (SSO-LeNet) was developed and evaluated on three benchmark datasets: MNIST, Fashion-MNIST and CIFAR-10. The proposed hybrid model was compared against the original LeNet as well as LeNet optimize by PSO. The study's findings revealed the superiority of SSO-LeNet over PSO-Le Net in terms of accuracy, while also outperforming the original LeNet in terms of both accuracy and time. In another study (Zhihua Cui et al., 2018), CNN was utilized for detecting malicious code variants, and the Bat Algorithm was employed to solve the imbalance of data among different malware families. The study's experiments demonstrated that the proposed model produced good accuracy and speed in solving the case of interest.

In contrast to the above methods that use optimization techniques to optimize machine learning parameters, a study in Dong et al. (2016) demonstrated a self-adaptive mechanism for extreme learning machine (SaELM). In the study, the proposed SaELM selects the best neuron number in hidden layer to form the Neural Networks without the need to tune any parameters during training process. Compared to conventional Back propagation Neural Network, SaELM showed good performance in solving classification tasks.

Concerning the above matters, this study proposed a hybrid predictive model for stock price prediction, based on the Barnacles Mating Optimizer (Sulaiman et al., 2020) with Artificial Neural Network (BMO-ANN). The BMO is a relatively new optimization algorithm that has demonstrated its effectiveness in solving optimization problems across fields, including engineering (Sulaiman et al., 2020), medical (Mustaffa & Sulaiman, 2021), biomedical engineering (Dutta et al., 2022), among others. BMO is an evolutionary-based algorithm inspired by the unique characteristics of barnacles in nature, which are hermaphroditic organisms with both male and female reproductive systems. One of the unique features of barnacles is their ability to extend their reproductive organs to several times their body length, up to seven to eight times.

As demonstrated in Sulaiman and Mustaffa (2022), the efficiency of BMO compared to other meta-heuristic algorithms have been proven through its competitive results against several algorithms, including PSO, Augmented Group Research Optimization (AGSO), and Imperialistic Competitive Algorithm (ICA). Similarly, in Peddakapu et al. (2022), besides the mentioned methods in Peddakapu et al. (2022), BMO was shown to outperform GA, Cuckoo Search (CS) and Bat Algorithm (BA) as well. This is due to its ability in escaping local optima, allowing global optima to be achieved. Moreover, algorithms like GA, PSO, ICA and Bacterial Foraging Optimization Algorithm (BFOA) were reported to exhibit slow convergence speed (Peddakapu et al., 2022). Other optimization algorithms, such as Artificial Bee Colony (ABC), lack local search ability and accuracy (Gu et al., 2022), while Runge-Kutta Optimizer (RUN) suffers from an imbalance between the exploitation and exploration phases (Nassef et al., 2022). In addition to its superiority, BMO also offers convenience in implementation, as it only has one control parameter, called penis length, *pl*, in addition to the number of population and maximum iteration. This hybrid predictive model is specifically designed for stock price prediction, which is inherently challenging due to high volatility in practice (Liu & Ma, 2022; Kumbure et al., 2022). The proposed predictive model is believed to be able to overcome the shortcomings of classical ANN in prediction task.

The remainder of this paper is organized as follows: In Section 2, a review of related literature is provided. Section 3 introduces the mathematical model of BMO, while Section 4 describes the ANN architecture used in the study. The implemented methodology is presented in Section 5, which includes information on the dataset used, the hybrid BMO-ANN, and the performance indices employed. Findings of the study are discussed in Section 6, while Section 7 provides a conclusion, followed by future directions for work.

2. Literature review

In literature, there is extensive discussion of the application of machine learning for time series prediction of stock prices. For instance, in Kumar Chandar (2021), researchers presented a method for intraday stock price prediction that hybrid PSO with Back Propagation Neural Network (BPNN). PSO was employed to automatically tune the BPNN's parameters for better accuracy. To determine the optimal training and testing set division, a series of experiments were conducted using ratios of 80:20, 70:30, 60:40 and 50:50, respectively. Upon completion of simulation tasks, the proposed PSO-BPNN consistently outperformed other methods in terms of error rates, as measured by the Root Mean Square Error (RMSE), Hit Rate (HR) and prediction accuracy.

An optimal Deep Learning-based Long Short-Term memory (LSTM) for stock price prediction using Twitter sentiment analysis was presented in Swathi et al. (2022). As in Kumar Chandar (2021), this method uses a hybrid approach that combines machine learning with an optimization technique to efficiently optimize the relevant parameters. In this case, Teaching and Learning Based Optimization (TLBO) was employed to tune the output unit of the LSTM, while Adam Optimizer is used to determine the learning rate. Findings of the study suggest the good performance of TLBO-LSTM over other identified methods in terms of predictive accuracy.

In Jing et al. (2021), researchers also investigated the impact of sentiment analysis on stock price prediction. The study integrated LSTM with Convolutional Neural Network (CNN) to analyze investor sentiments before integrating them with technical indicators as inputs for predicting one-day-ahead closing prices. For this matter, a sentiment analysis model based on CNN was trained and tested. The LSTM-CNN model produced lower Mean Absolute Percentage Error (MAPE) than both single model and other prediction models proposed in previous work, demonstrating its superiority.

In Vijh et al. (2020), a study compared the effectiveness of ANN and Random Forest (RF) for stock price prediction. Based on several statistical criteria, the results favored ANN. Similarly, in Liu and Ma (2022), an improved ANN based on Elman Neural Network and quantum mechanics (QENN) was proposed for stock price prediction, and findings of the study proved that the QENN has advantages over classical ANN. Another study in Chhajer et al. (2022) presented an approach for stock price prediction using ANN, Support Vector Machines (SVM) and Long Short Term Memory (LSTM).

A study in Liu et al. (2022) presented the use of Deep Learning Neural Network (DLNN) for stock price prediction in a case study conducted in China. The performance of DLNN was compared against single-layer model, demonstrating the superiority of DLNNs for the case study at hand. A similar approach can be seen in Wei et al. (2022), where a DL based on LSTM was employed to predict stock closing prices. The proposed method was tested on two datasets, namely CSI 300 and Hang Seng Index, the DL exhibits good performance by producing lower error rates.

The employment of machine learning extends beyond predicting stock prices. In a recent study (Khanduzi & Sangaiah, 2023), researchers employed a Recurrent Neural Network (RNN)-based bilevel programming problem formulation to tackle the problem of continuous defensive location problem (CDLP). By using the RNN, they were able to optimize the placement of defense facilities in the CDLP in a large size example while also reducing computational costs. The study found out that the proposed approach outperformed other methods that were tested. Another employment of RNN also can be seen in Sumathi et al. (2022). In the study, the researchers explored the use of a RNN and ZigBee protocol in an IoT environment. Findings of the study demonstrated that the proposed method significantly improved the network lifetime and decreased energy consumption.

Regarding the importance of personalized recommendation, a study in Sangaiah et al. (2023) developed a medical recommendation system based on two community detection algorithms. The algorithms were applied to a graph of users and physicians to create the proposed recommender system. The study used 80,000 records for training and 20,000 for testing. Prior to training and testing, all datasets were normalized based on Min Max Normalization. Upon completion of the experiment, the proposed recommender system was found to produce highly accurate recommendations compared to those reported in the literature.

3. Optimization based on Barnacles Mating Optimizer

This section provides the mathematical model of BMO.

3.1. Phase 1: initialization

Initially, the candidate solutions are represented as shown in the following equation. This stage involves the generation of candidate solutions which are randomly generated. An evaluation will be executed to obtain the initial performance of candidates, followed by the sorting phase:

$$X = \begin{bmatrix} x_l^1 & \cdots & x_1^N \\ \cdots & \cdots & \cdots \\ x_n^l & \cdots & x_n^N \end{bmatrix} \tag{1}$$

where N indicates the dimension of control variables to be optimized while n represents the population of barnacles which can be considered as the candidate for the solution. The control variables in (1) is subject to the boundaries of the problem at hands, as defined in the next two equations, respectively:

$$ub = \begin{bmatrix} ub_1, & \ldots, & ub_i \end{bmatrix} \tag{2}$$

$$lb = \begin{bmatrix} lb_1, & \ldots, & lb_i \end{bmatrix} \tag{3}$$

where ub is the upper bound and lb is lower bound of i th variables.

3.2. Phase 2: selection

The selection process for mating to produce new offspring is performed randomly. However, it is subject to a parameter that need to be tuned namely pl, which represents the range of the barnacles that can mate. Barnacles are classified as hermaphroditic organism that can contribute to and receive sperm from others (Nassef et al., 2022). Nevertheless, for the simplicity in the formulation of the BMO algorithm, it is assumed that at one particular time, each barnacle can be fertilized by only one barnacle. In nature, the self-mating can occur but is very rare, as mentioned in Liu & Ma (2022). Therefore, it will not be considered in the mathematical model of BMO.

3.3. Phase 3: reproduction

In BMO, the reproduction of new offspring is based on the Hardy-Weinberg principle according to which in the new offspring created the features or characters from the parents are inherited, as:

$$x_i^{N_{new}} = px_{barnacles}^N + qx_{banacle_m}^N \tag{4}$$

where p indicates the normally distributed pseudo random numbers between the range of [0, 1], $q = 1 - p$, $x_{barnacles\_d}^N$ and $x_{barnacle\_m}^N$ represent the variable of Dad and Mum of barnacles, respectively. It can be said that p and q represent the proportion of characteristics inherited from the Dad and Mum, respectively, in generating new offspring. The offspring inherits the features from both parents based on a probability of a random number between 0 and 1. For example, if p if 0.55 (randomly generated), it means that the new offspring inherits 55% of the Dad's features and 45% from the Mum's features. It also can be noted that (4) can be seen as exploitation process of the BMO for obtaining new solutions. The exploitation and exploration processes in BMO are highly dependent on the value of pl. If the selection of barnacles to be mated falls within the range of barnacles selected from the Dad and Mum, exploitation occurs and (4) is applied. Sperm cast occurs when the selection of barnacles to be mated exceeds the value of pl set initially. The sperm cast process can be defined as follows:

$$x_i^{nnew} = rand() \times x_{barnacleM}^N \tag{5}$$

where $rand()$ denotes the random number generated between the range of $0 \sim 1$. The pseudo code of BMO is given in Fig. 1.

4. Regression based on Artificial Neural Network

ANN is one of the notable machine learning techniques that has the capability of simulating and analysing complex patterns in unstructured data. For a prediction tasks, a Multilayer Perceptron (MLP) architecture is employed. MLP is a feedforward, supervised learning network consisting of 3 layers: an input layer, a hidden layer and an output layer. According to Fig. 2, the input layer consists of six inputs, namely the difference of high and low prices (*HL*), closing and opening prices (*CO*), moving averages of 7 (*MA7*), 14 (*MA14*) and 21 (*MA21*) days, as well as the standard deviation of stock prices for the past seven days (*Std7)*, as suggested in (Vijh et al., 2020). Meanwhile, the output is the next day's closing price (*yCP*). Instead of using the Back Propagation (BP) algorithm to train the network, this paper proposes an optimization approach mentioned in Section 2, BMO to optimize the weights and biases for training the developed network.

According to Fig. 2, the input layer receives input signals from the source and transmits them to every hidden neuron within the hidden layer. Meanwhile, the inputs are multiplied with the weight, w_{ji} and summed together with the bias to become the input, u for the hidden neurons, which are located in the hidden layer. The activation function used in the hidden neurons is the sigmoid function, which is expressed as follows:

$$O_j = \frac{1}{1 + e^{-u}} \tag{6}$$

where O_j is the output from each hidden neuron. Then, the outputs from each hidden neurons are multiplied with the weights connected between hidden and output layers, w_{kj} and summed together with the bias at output neuron to become the input for the output neuron. The output is normally determined using the linear function, and in this paper, the output is the next day's of closing stock price.

5. Methodology

This section is dedicated to describing the implemented methodology for this study. This includes a description on the dataset, variable derivation, hybrid model of BMO-ANN, and finally, the evaluation criteria.

5.1. Data description

In this study, the daily historical dataset of Yahoo stock from November 2015 until November 2020 is employed. The dataset consists of information about the stock, including High, Low, Open, Close, Volume and Adjacent Close, which can be retrieved from the Kaggle.com website (Time Series Forecasting with Yahoo Stock Price, 2022). A sample of the dataset is tabulated in Table 1.

5.2. Variable derivation

For experimental purposes, six variables were derived from the original input, namely the difference of *HL, CO, MA7, MA14* and *MA21* days, as well as *Std7*. These inputs were fed to the prediction model to predict the next day's closing price.

Algorithm 1: BMO

```
1.  Initialize the population of barnacles:
    Xi = rand(n,2)
2.  Calculate the fitness of each barnacle:
    Fitness=calculate_fitness(Xi)
3.  Sorting to locate the best result at the top of the population (T=the best solution):
    [sorted_fitness, idx]=sort(fitness)
    T=Xi(idx(1),:)
4.  while (I<Maximum iterations):
    for I=1:max_iter
5.      Set the value of pl
6.      Selection based on the following equations:
        barnacle_d=randperm(n)
        barnacle_m=randperm(n)
7.      if selection of Dad and Mum=pl:
        if rand() <=pl
8.          for each variable:
            for i=1:size(Xi,2)
9.              Off-spring generation using (4)
                p=rand()
                q=1-p
                X_i^{N_new} = p x^N_{barnacles_d} + q x^N_{barnacle_m}
10.         end for
11.     else if selection Dad and Mum>pl
12.         For each variable
            for i=1:size(Xi,2)
13.             Off-spring generation using equation (5)
                x_i^{n_new}=rand() × x^N_{barnacle_m}
14.         end for
15.     end if
16.     Bring the current barnacle back if it is out of boundaries
        for i=1:size(Xi_new,1)
            for j=1:size(Xi_new,2)
            if Xi_new(i,j)<0
              Xi_new(i,j)=0
            else if Xi_new(i,j) >1
              Xi_new(i,j)=1
            end if
        end for
17.     Calculate the fitness of each barnacle
        Fitness_new=calculate_fitness(Xi_new)
18.     Sorting and update T if there is a better solution
        if sorted_fitness_new(1) < sorted_fitness(1)
           T=Xi_new(idx_new(1),:)
        end
        Update Xi and fitness
        Xi=Xi_new
        Fitness=fitness_new
        end
    end
19.     I= I+1
20. end while
21. Return T
```

Fig. 1. BMO algorithm.

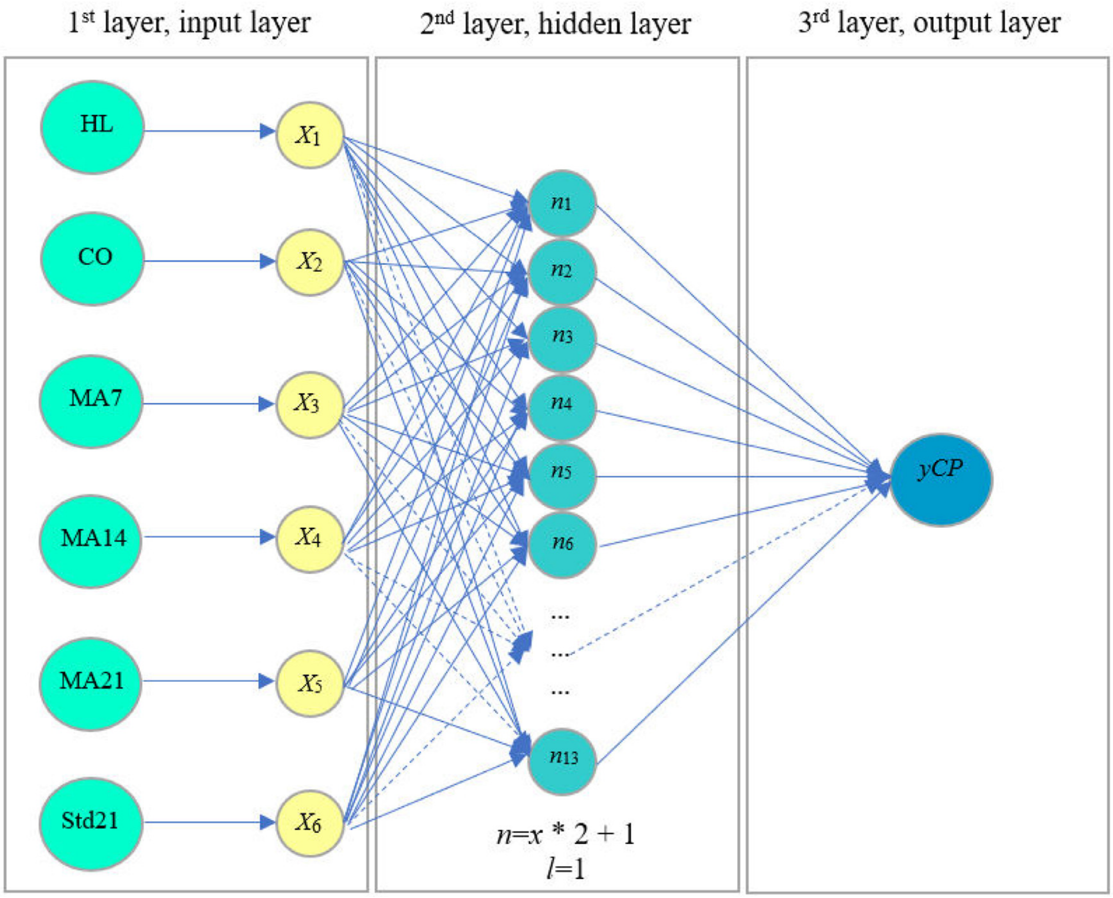

Fig. 2. MLP architecture for stock price prediction.

Table 1
Sample of dataset.

Date	High	Low	Open	Close	Volume	Adjacent Close
1/1/2016	2062.54	2043.62	2060.59	2043.94	2,655,330,000	2043.9399
1/2/2016	2062.54	2043.62	2060.59	2043.94	2,655,330,000	2043.9399
1/3/2016	2062.54	2043.62	2060.59	2043.94	2,655,330,000	2043.9399
1/4/2016	2038.2	1989.68	2038.2	2012.66	4,304,880,000	2012.6600
1/5/2016	2021.94	2004.17	2013.78	2016.71	3,706,620,000	2016.7099
1/6/2016	2011.71	1979.05	2011.71	1990.26	4,336,660,000	1990.2600
1/7/2016	1985.32	1938.83	1985.32	1943.09	5,076,590,000	1943.0899
1/8/2016	1960.40	1918.46	1945.97	1922.03	4,664,940,000	1922.0300
1/9/2016	1960.40	1918.46	1945.97	1922.03	4,664,940,000	1922.0300
1/10/2016	1960.40	1918.46	1945.97	1922.03	4,664,940,000	1922.0300
1/11/2016	1935.65	1901.1	1926.12	1923.67	4,607,290,000	1923.6700
1/12/2016	1947.38	1914.35	1927.83	1938.68	4,887,260,000	1938.6800
1/13/2016	1950.33	1886.41	1940.34	1890.28	5,087,030,000	1890.2800
1/14/2016	1934.47	1878.93	1891.68	1921.84	5,241,110,000	1921.8399

5.3. Hybrid Barnacle Mating Optimizer-Artificial Neural Network

To get the optimized weights and biases for the ANN model, BMO is employed in this study. The BMO function is embedded in the ANN, where BMO will run until the maximum iteration is reached (in this study, it is set to 100). In this paper, the variables to be optimized are the weights in the network, viz. w_{ji} and w_{kj} together with the biases at hidden and output layers. The total number of hidden neurons are set to 11 based on ($2^*n + 1$), where n is the number of inputs, which in this case is 6. Thus, the total variables to be optimized are 105 (6 inputs*11 w_{ji} + 11 biases at hidden neurons +11 w_{kj} +1 bias at output neuron), and the objective function is the normalized MSE. The flowchart of BMO-ANN is visualized in Fig. 3.

5.4. Evaluation criteria

For evaluation purposes, two statistical metrics were employed, namely Mean Square Error (MSE) and Root Mean Square Percentage Error (RMSPE). The definitions of both metrics are as follows:

$$MSE = \frac{1}{N}\left(y_n - \widehat{y_n}\right)^2 \tag{7}$$

$$RMSPE = \sqrt{\frac{\sum_{n=1}^{N}\left(\frac{y_n - \widehat{y_n}}{y_n}\right)^2}{N}} \tag{8}$$

where $n = 1, 2, \ldots, N$; N is the number of samples, y_n is the actual value of the nth sample, and $\widehat{y}_n$ is the predicted value of the nth sample. MSE

```
Algorithm 2: BMO-ANN
Load dataset, yahoo_stock_price.csv
Training-testing configuration
Training data, Tr=0.6, Testing data, Ts=0.4
Number of hidden neurons, numHidNeu=13
Create a neural network, net = feedforward (n)
Configure neural network for yahoo_stock.csv
Create handle to the MSE_TEST function, that calculates MSE
   While (i<=Maximum iteration)
      Call BMO to optimize ANN weights and biases (See Fig. 1)
      Calculate MSE
      If (current_MSE < old_MSE)
         MSE=current_MSE
      Else
         MSE=old_MSE
   End while
Save network BMO_ANN
Test the trained network and evaluate the results
```

Fig. 3. BMO-ANN.

represents the average mean square error between the predicted value and the actual value, while RMSPE is the square root of the mean of the square of all the errors, expressed in percentage. Thus, the smaller the metric, the better the prediction.

The main computational cost of the algorithm is the calculation of the fitness of the objective function, which needs to be evaluated for each barnacle in the population at each iteration. The fitness function evaluation involves computing the objective function, which minimizes the errors (NMSE) from the tuned weights and biases of FFNN. The time complexity of the algorithm can be expressed as $O(Max\_Iter * n * f)$, where *Max_Iter* is the maximum number of iterations, n is the population size, and f is the time complexity of the fitness function evaluation. The space complexity of the algorithm is $O(n * num\_variables)$, where *num_variables* is the number of weights and biases of FFNN.

6. Results and discussion

To determine the optimal data division for stock price prediction using BMO-ANN, the proposed model was experimented with three data divisions for training and testing, namely 60:40, 70:30 and 80:20. The results obtained from these tests are presented in Table 2.

Based on the results presented in Table 2, the BMO-ANN model produced the lowest MSE and RMSPE values when using a data division of 60:40 for training and testing, with recorded values of 0.6619 and 6.5538%, respectively. In comparison, using a data division of 70:30 led to higher error rates, with an MSE of 1.6467 and RMSPE of 9.1282%. Lower error rates were observed when using a data division of 80:20 compared to the 70:30 split, with an MSE of 1.0197 and RMSPE of 7.9237.

Table 2
BMO-ANN for stock price prediction using different data division.

Metrics/Data Division	60:40	70:30	80:20
MSE	**0.6619**	1.6467	1.0197
RMSPE(%)	**6.5538**	9.1282	7.9237

Table 3
Stock price prediction: BMO-ANN vs. SSA-ANN vs. WOA-ANN vs. MFO-ANN vs. GWO-ANN.

	MSE	RMSPE(%)
BMO-ANN	0.6619	4.5617
SSA-ANN	4.5617	17.7095
WOA-ANN	1.0175	8.0778
MFO-ANN	15.4502	36.1689
GWO-ANN	2.7742	13.7597

Based on the 60:40 data division, experiments were conducted to compare BMO-ANN with four other identified hybrid algorithms using ANN: Salp-Swarm Algorithm (SSA-ANN), Whale Optimization Algorithm (WOA-ANN), Moth Flame Optimizer (MFO-ANN) and grey Wolf Optimizer (GWO-ANN).

Based on the data presented in Table 3, it is learned that the BMO-ANN outperformed the other four identified algorithms in terms of having the lowest MSE and RMSPE values, with values of 0.6619 and 4.5617%, respectively. The WOA-ANN had the second-best performance, with MSE and RMSPE values of 1.0175 and 8.0778%, re-

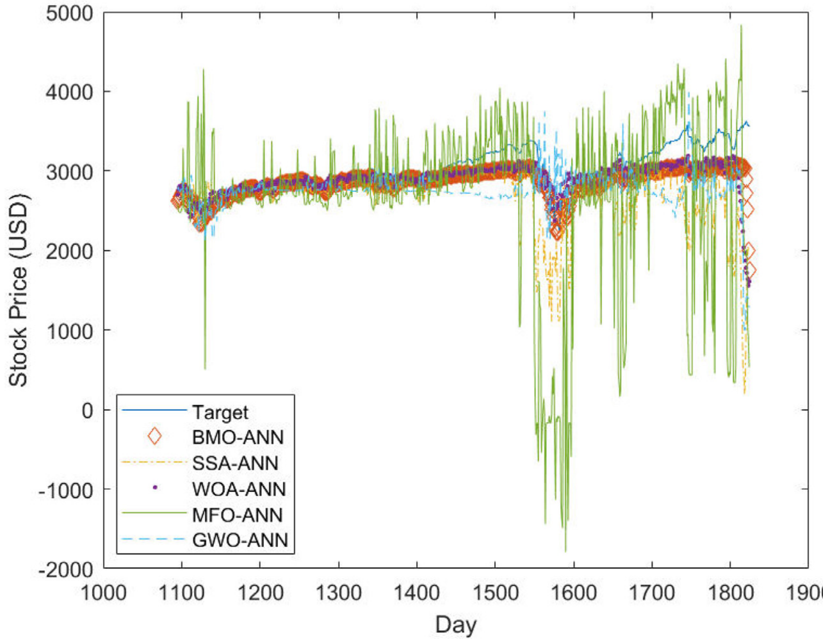

Fig. 4. Comparison graph for stock price prediction: BMO-ANN vs. SSA-ANN vs. WOA-ANN.

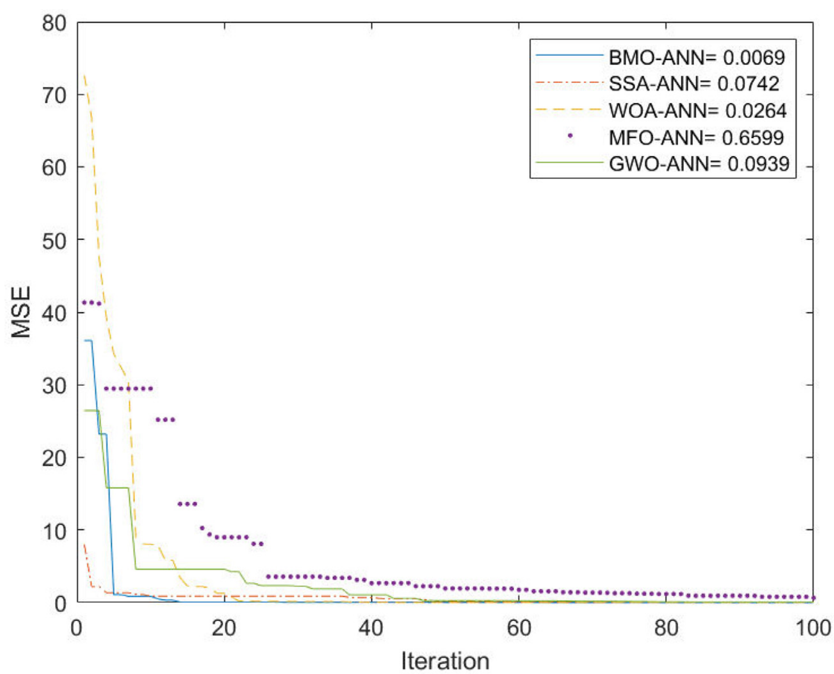

Fig. 5. Convergence graph: BMO-ANN vs. SSA-ANN vs. WOA-ANN vs. MFO-ANN vs. GWO-ANN.

spectively, while the GWO-ANN ranked third with values of 2.7742 of MSE and 13.7597% RMSPE. In contrast, the SSA-ANN yielded higher error rates than those of the GWO-ANN, with MSE and RMSPE values of 4.5617 and 17.7095%, respectively. The MFO-ANN had the highest error rates, generating MSE and RMSPE values of 15.4502 and 36.1689%, respectively. Overall, the data from Table 3 indicates that the BMO-ANN algorithm had the best performance among the five algorithms, while the MFO-ANN had the worst performance. The WOA-ANN and GWO-ANN had the intermediate performance, with the SSA-ANN algorithm having the second-worst performance.

Fig. 4 provides a visual representation of the results, with the target values shown as a straight blue line and the predictions made by each algorithm indicated by different colored markers and lines. Specifically, the BMO-ANN results are represented by red diamonds, the SSA-ANN results by dash-dot orange lines, the WOA-ANN results by purple dots, the MFO-ANN results by a straight green line, the GWO-ANN results by a straight green line. From the figure, it is evident that the MFOA-ANN not only failed to produce accurate predictions but also did not follow the actual pattern, particularly between day 1500 and 1800, resulting in higher error rates. In contrast, the BMO-ANN showed good generalization ability and produced predictions that closely matched the target values, even during sudden drops in prices around day 1600.

Fig. 5 illustrates the convergence rates of the five identified algorithms. The results indicate that the proposed BMO-ANN achieved the

Table 4
Significant test for stock price prediction: BMO-ANN vs. SSA-ANN vs. WOA-ANN vs. MFO-ANN vs. GWO-ANN.

Methods	Sig. (2-tailed)
BMO-ANN vs. SSA-ANN	0.0000
BMO-ANN vs. WOA-ANN	0.0000
BMO-ANN vs. MFO-ANN	0.0000
BMO-ANN vs. GWO-ANN	0.0000

lowest convergence rate, with an MSE value of 0.0069 (indicated by straight blue line). The WOA-ANN followed closely, recording an MSE of 0.0264 of MSE (denoted by dashed yellow line). The SSA-ANN ranked third with an MSE of 0.0742, closely followed by GWO-ANN, which recorded an MSE of 0.0939. In contrast, the MFO-ANN had the highest convergence rates, with an MSE of 0.6599, which was significantly higher than the other algorithms. The high convergence rate of MFO-ANN suggests that the algorithm may have been stuck in local optima, rather than global optima, resulting in poor performance in the prediction task.

Additionally, a paired sample T-test was conducted to evaluate the performance of the proposed BMO-ANN. As shown in Table 4, the results indicate a significant difference in the means between BMO-ANN and the other identified hybrid algorithms, with a significance level of 0.05%.

7. Conclusion

In this study, a hybrid predictive model for stock price based on BMO-ANN was proposed and evaluated on a stock price time series dataset. The BMO was utilized to optimize the weight and bias of ANN for generalization in prediction, and the results show that the proposed model outperforms other identified hybrid algorithms based on two evaluation indices, MSE and RMSPE. Paired sample T-test results also confirmed the significance level of the difference in means between BMO-ANN and other identified hybrid algorithms. Therefore, it is evidence that the BMO-ANN is efficient for the time series data of interest.

In future work, the study will consider the speed of the convergence and compare it against other competitive methods which includes Monarch Butterfly Optimization (MBO), Earthworm Optimization Algorithm (EWA), Elephant Herding Optimization (EHO), Moth Search (MS) algorithm, Slime Mould Algorithm (SMA), Hunger Games Search (HGS), Runge Kutta Optimizer (RUN), Colony Predation Algorithm (CPA), Harris Hawks Optimization (HHO), among others. Additionally, the BMO-FNN can be tested on other time series datasets to evaluate its effectiveness in different scenarios.

Declaration of Competing Interest

The authors declare that there is no conflict of interest.

Acknowledgments

This work is financially supported by Universiti Malaysia Pahang Research grant (RDU220379)

References

Wang, P., Zheng, X., Li, J., & Zhu, B. (2020). Prediction of epidemic trends in COVID-19 with logistic model and machine learning technics. *Chaos, Solitons & Fractals, 139*, Article 110058. https://doi.org/10.1016/j.chaos.2020.110058.

Portillo Juan, N., & Negro Valdecantos, V. (2022). Review of the application of Artificial Neural Networks in ocean engineering. *Ocean Engineering, 259*, Article 111947. https://doi.org/10.1016/j.oceaneng.2022.111947.

Vapnik, V. N. (1995). *The nature of statistical learning theory* (2nd ed.). New York: Springer-Verlag.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least squares support vector machines*. Leuven, Belgium: World Scientific Publishing Co. Pte. Ltd..

Lalmuanawma, S., Hussain, J., & Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons & Fractals, 139*, Article 110059. https://doi.org/10.1016/j.chaos.2020.110059.

Shahid, F., Zameer, A., & Muneeb, M. (2020). Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. *Chaos, Solitons & Fractals, 140*, Article 110212. https://doi.org/10.1016/j.chaos.2020.110212.

Agrawal, P., Katal, N., & Hooda, N. (2022). Segmentation and classification of brain tumor using 3D-UNet deep Neural Networks. *International Journal of Cognitive Computing in Engineering, 3*, 199–210. https://doi.org/10.1016/j.ijcce.2022.11.001.

Solayman, S., Aumi, S. A., Mery, C. S., Mubassir, M., & Khan, R. (2023). Automatic COVID-19 prediction using explainable machine learning techniques. *International Journal of Cognitive Computing in Engineering, 4*, 36–46. https://doi.org/10.1016/j.ijcce.2023.01.003.

Yan, Y., Wang, X., Ren, F., Shao, Z., & Tian, C. (2022). Wind speed prediction using a hybrid model of EEMD and LSTM considering seasonal features. *Energy Reports, 8*, 8965–8980. https://doi.org/10.1016/j.egyr.2022.07.007.

Shen, Z., Fan, X., Zhang, L., & Yu, H. (2022). Wind speed prediction of unmanned sailboat based on CNN and LSTM hybrid Neural Network. *Ocean Engineering, 254*, Article 111352. https://doi.org/10.1016/j.oceaneng.2022.111352.

Kara, Y., Boyacioglu, M. A., & Baykan, O. K. (2011). Predicting direction of stock price index movement using Artificial Neural Networks and support vector machines: the sample of the Istanbul stock exchange. *Expert Systems with Applications, 38*(2011), 5311–5319.

Saxena, N., & Varshney, D. (2021). Smart home security solutions using facial authentication and speaker recognition through Artificial Neural Networks. *International Journal of Cognitive Computing in Engineering, 2*, 154–164. https://doi.org/10.1016/j.ijcce.2021.10.001.

Zhihua Cui, J. C., Xue, F., Cai, X., Cao, Y., & Wang, G. G. (2018). Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics, 14*(7), 3187–3196.

Öztunç Kaymak, Ö., & Kaymak, Y. (2022). Prediction of crude oil prices in COVID-19 outbreak using real data. *Chaos, Solitons & Fractals, 158*, Article 111990. https://doi.org/10.1016/j.chaos.2022.111990.

Liu, T., Ma, X., Li, S., Li, X., & Zhang, C. (2022a). A stock price prediction method based on meta-learning and variational mode decomposition. *Knowledge-Based Systems, 252*, Article 109324. https://doi.org/10.1016/j.knosys.2022.109324.

Swaraj, A., Verma, K., Kaur, A., Singh, G., Kumar, A., & de Sales, L. M. (2021). Implementation of stacking based ARIMA model for prediction of Covid-19 cases in India. *Journal of Biomedical Informatics, 121*, 1–11. https://doi.org/10.1016/j.jbi.2021.103887.

Lai, K. K., Yu, L., Wang, S., & Huang, W. (2006). Hybridizing exponential smoothing and Neural Network for financial time series predication. In V. N. Alexandrov, G. D. van Albada, P. M. A. Sloot, & J. Dongarra (Eds.). In *Computational Science ICCS 2006. ICCS 2006. Lecture Notes in Computer Science: vol. 3994*. Heidelberg: Springer, Berlin. https://doi.org/10.1007/11758549_69.

Naderi, M., Khamehchi, E., & Karimi, B. (2019). Novel statistical forecasting models for crude oil price, gas price, and interest rate based on meta-heuristic bat algorithm. *Journal of Petroleum Science and Engineering, 172*, 13–22. https://doi.org/10.1016/j.petrol.2018.09.031.

Karasu, S., & Altan, A. (2022). Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization. *Energy, 242*, Article 122964. https://doi.org/10.1016/j.energy.2021.122964.

Tian, Z. (2020). Short-term wind speed prediction based on LMD and improve FA optimized combined kernel function LSSVM. *Engineering Applications of Artificial Intelligence, 91*, 1–24.

Yeh, W. C., Lin, Y. P., Liang, Y. C., Lai, C. M., & Huang, C. L. (2023). Simplified swarm optimization for hyperparameters of convolutional Neural Networks. *Computers & Industrial Engineering, 177*, Article 109076. https://doi.org/10.1016/j.cie.2023.109076.

Dong, Y. Q., Zhao, X. J., Wang, G. G., & Lu, M. (2016). Self-adaptive extreme learning machine. *Neural Computing and Applications, 27*, 291–303.

Sulaiman, M. H., Mustaffa, Z., Saari, M. M., & Daniyal, H. (2020). Barnacles Mating Optimizer: A new bio-inspired algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 87*, Article 103330. https://doi.org/10.1016/j.engappai.2019.103330.

Mustaffa, Z., & Sulaiman, M. H. (2021). COVID-19 confirmed cases prediction in China based on Barnacles Mating Optimizer-least squares support vector machines. *Cybernetics and Information Technologies, 21*(4), 62–76. https://doi.org/10.2478/cait-2021-0043.

Dutta, A. K., Mageswari, R. U., Gayathri, A., Dallfin Bruxella, J. M., Ishak, M. K., Mostafa, S. M., Hamam, H., et al., (2022). Barnacles Mating Optimizer with deep transfer learning enabled biomedical malaria parasite detection and classification. *Computational Intelligence and Neuroscience, 2022*, Article 7776319. https://doi.org/10.1155/2022/7776319.

Sulaiman, M. H., & Mustaffa, Z. (2022). Optimal chiller loading solution for energy conservation using Barnacles Mating Optimizer algorithm. *Results Control and Optimization, 7*, Article 100109. https://doi.org/10.1016/j.rico.2022.100109.

Peddakapu, K., Srinivasarao, P., Mohamed, M. R., Arya, Y., & Krishna Kishore, D. J. (2022). Stabilization of frequency in Multi-Microgrid system using Barnacle Mating Optimizer-based cascade controllers. *Sustainable Energy Technologies and Assessments, 54*, Article 102823. https://doi.org/10.1016/j.seta.2022.102823.

Gu, Z., Zhu, M., Cui, Q., Qiao, M., Zhu, W., Zhu, Q., Li, L., & Han, M. (2022). Research on broadband equivalent modeling for the dielectric response of transformer bushings based on the improved artificial bee colony algorithm. *Energy Reports, 8*, 366–372. https://doi.org/10.1016/j.egyr.2022.03.094.

Nassef, A. M., Houssein, E. H., Helmy, B. E., Fathy, A., Alghaythi, M. L., & Rezk, H. (2022). Optimal reconfiguration strategy based on modified Runge Kutta optimizer to mitigate partial shading condition in photovoltaic systems. *Energy Reports, 8*, 7242–7262. https://doi.org/10.1016/j.egyr.2022.05.231.

Liu, G., & Ma, W. (2022). A quantum Artificial Neural Network for stock closing price prediction. *Information Sciences, 598*, 75–85. https://doi.org/10.1016/j.ins.2022.03.064.

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications, 197*, Article 116659. https://doi.org/10.1016/j.eswa.2022.116659.

Kumar Chandar, S. (2021). Hybrid models for intraday stock price forecasting based on Artificial Neural Networks and metaheuristic algorithms. *Pattern Recognition Letters, 147*, 124–133. https://doi.org/10.1016/j.patrec.2021.03.030.

Swathi, T., Kasiviswanath, N., & Rao, A. A. (2022). An optimal deep learning-based LSTM for stock price prediction using twitter sentiment analysis. *Applied Intelligence, 52*(12), 13675–13688. https://doi.org/10.1007/s10489-022-03175-2.

Jing, N., Wu, Z., & Wang, H. (2021). A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. *Expert Systems with Applications, 178*, Article 115019. https://doi.org/10.1016/j.eswa.2021.115019.

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia Computer Science, 167*, 599–606. https://doi.org/10.1016/j.procs.2020.03.326.

Chhajer, P., Shah, M., & Kshirsagar, A. (2022). The applications of Artificial Neural Networks, support vector machines, and long–short term memory for stock market prediction. *Decision Analytivcs Journal, 2*, Article 100015. https://doi.org/10.1016/j.dajour.2021.100015.

Liu, Q., Tao, Z., Tse, Y., & Wang, C. (2022b). Stock market prediction with deep learning: The case of China. *Finance Research Letters, 46*, Article 102209. https://doi.org/10.1016/j.frl.2021.102209.

Wei, J., Xu, Q., & He, C. (2022). Deep learning of predicting closing price through historical adjustment closing price. *Procedia Computer Science, 202*, 379–384. https://doi.org/10.1016/j.procs.2022.04.052.

Khanduzi, R., & Sangaiah, A. K. (2023). An efficient recurrent Neural Network for defensive Stackelberg game. *Journal of Computational Science, 67*, Article 101970. https://doi.org/10.1016/j.jocs.2023.101970.

Sumathi, A. C., Javadpour, A., Pinto, P., Sangaiah, A. K., Zhang, W., & Khaniabadi, S. M. (2022). NEWTR: A multipath routing for next hop destination in internet of things with artificial recurrent Neural Network (RNN. *International Journal of Machine Learning and Cybernetics, 13*, 2869–2889.

Sangaiah, A. K., Rezaei, S., Javadpour, A., & Zhang, W. (2023). Explainable AI in big data intelligence of community detection for digitalization e-healthcare services. *Applied Soft Computing, 136*, Article 110119. https://doi.org/10.1016/j.asoc.2023.110119.

"Time Series Forecasting with Yahoo Stock Price." https://www.kaggle.com/datasets/arashnic/time-series-forecasting-with-yahoo-stock-price. 2022