Franklin Open 8 (2024) 100137

Contents lists available at ScienceDirect

Franklin Open

journal homepage: www.elsevier.com/locate/fraope

Improving Earth surface temperature forecasting through the optimization of deep learning hyper-parameters using Barnacles Mating Optimizer

Zuriani Mustaffa [a,*], Mohd Herwan Sulaiman [b], Muhammad 'Arif Mohamad [a]

[a] *Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Pahang 26600, Malaysia*
[b] *Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, Pahang 26600, Malaysia*

ARTICLE INFO

Keywords:
Barnacles Mating Optimizer
Deep learning
Optimization
Time series prediction
Earth Surface Temperature
Climate change

ABSTRACT

Time series forecasting is crucial across various sectors, aiding stakeholders in making informed decisions, planning for the short and long term, managing risks, optimizing profits, and ensuring safety. One significant application of time series forecasting is predicting Earth surface temperatures, which is vital for civil and environmental sectors such as agriculture, energy, and meteorology. This study proposes a hybrid forecasting model for Earth surface temperature using Deep Learning (DL). To improve the DL model's performance, an optimization algorithm called Barnacles Mating Optimizer (BMO) is integrated to optimize both weights and biases. The forecasting model is trained on a global temperature dataset with seven inputs and compared with DL models optimized by Particle Swarm Optimization (PSO), Harmony Search Algorithm (HSA), and Ant Colony Optimization (ACO). Additionally, a comparison is made with the Autoregressive Moving Average (ARIMA) method. Evaluation using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the coefficient of determination (R2) demonstrates the superior performance of DL optimized by BMO, showing minimal errors.

1. Introduction

Time series forecasting is vital and serves various purposes across different domains, including climate monitoring [1,2], irrigation management [3], urban planning [4–6], agriculture [7–12], infrastructure management [13,14], energy [15–18], and meteorology [19]. Reliable time series forecasting is of paramount importance as it empowers various stakeholders in different sectors. For example, in the context of climate monitoring, accurate forecasts enable governments and environmental agencies to anticipate extreme weather events, formulate disaster preparedness strategies, and implement climate change mitigation measures effectively. In urban planning, dependable forecasting aids city officials and urban designers in making informed decisions about infrastructure development, traffic management and other related tasks. Meanwhile, in the agriculture sectors, it helps optimize resource allocation, reduce waste, and enhance productivity. In meteorology, reliable forecasting not only ensures public safety by providing early warnings of severe weather conditions but also aids in aviation, maritime navigation, and tourism planning.

Time series forecasting of Earth surface temperature is critical in all these stated areas. It plays a pivotal role in tracking and responding to shifts in global climate patterns. Additionally, it ensures that cities can adapt to changing climate conditions and prioritize sustainable growth. Farmers can plan planting and harvesting schedules based on weather forecasts, leading to more efficient and sustainable agricultural practices. Accurate temperatures forecasts play a crucial role in these industries, contributing to safe and efficient operations.

Previously, statistical techniques like Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing were commonly used for time series forecasting. However, these techniques have limitations in capturing non-linear features that are present in many real-time situations [20,21]. As a result, their performance has diminished. The rise of machine learning methods, including Artificial Neural Networks (ANN), Support Vector Machines (SVM) [22], and Deep Learning (DL), has demonstrated encouraging outcomes in overcoming these challenges [23,24]. Machine learning models have demonstrated exceptional predictive accuracy in forecasting including Earth's temperature by effectively capturing intricate nonlinear relationships between air pollutant levels and various predictors, including meteorological conditions, and land use patterns [25]. This is substantiated by a multitude of published studies showcasing successful results.

Study in [26] demonstrated an efficient approach utilizing DL. The

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

https://doi.org/10.1016/j.fraope.2024.100137
Received 18 February 2024; Received in revised form 10 June 2024; Accepted 11 July 2024
Available online 14 July 2024

proposed approach, namely Propagated HIerarchical Learning Network (PHILNet), aiming to enhance the efficiency of neural networks in univariate multi-step time series forecasting, particularly focusing on reducing execution time. The presented approach yielded favorable outcomes, demonstrating a 35 % enhancement in mean squared error and a 2.6-fold reduction in training time when compared to the top-performing models across various time series. In [27], a hybrid of Twin Support Vector Regressor (TSVR), Singular Spectrum Analysis (SSA) and Grey Wolf Optimizer (GWO) was demonstrated for time series prediction of streamflow of hydropower reservoir. In the suggested investigation, the Grey Wolf Optimizer (GWO) was employed to optimize Time Series Vector Regression (TSVR), and the Singular Spectrum Analysis (SSA) was utilized as a data preprocessing tool for data identification. The superiority of SSA-GWO-TSVR over other methods was evident in the comparison. Another employment of TSVR in time series prediction also can be seen in [28].

The study referenced in [29] demonstrates the effectiveness of DL in improving the prediction of water quality parameters across various aquatic systems. Analysis of real-world data collected from Swan Canning Estuary sites reveals that the model is capable of predicting an increased number of hours with high scores, even when confronted with varying sizes of training and testing sets. Meanwhile, the surface water temperature of a lake was analyzed using a combination of Stacking Multilayer Perceptron and Random Forest (MLP-RF)) [30]. Hyperparameter tuning was conducted using Bayesian Optimization (BO) [31]. The related works on hybrid metaheuristic-deep learning also has been proposed in [32] for state of charge estimation problem. In comparison to other models, including a shallow multilayer perceptron neural network, a model integrating wavelet transform and multilayer perceptron neural network, as well as non-linear regression and air2-water models, the stacked MLP-RF model exhibited excellent forecasting capabilities across all lakes and forecast horizons.

Study in [33] proposed a hybrid Genetic Algorithm (GA) with DL to predict the PM2.5 dataset, including air pollutants and meteorological features in Istanbul metropolitan. GA is employed to find the best parameter combination for learning and dropout rate, the number of hidden layers and units in each hidden layer, activation function, loss function, and optimizer. Compared against DL with default hyperparameters and random search algorithms, the yielded results were in favor to DL optimized by GA. to confirm the efficacy of the genetic algorithm approach. Meanwhile, another hybrid GA is reported in [34] where in the study, an automated Hyperparameter Optimization (HPO) method based on a Parallel Genetic Algorithm (PGA) is proposed. The HPO process is divided into stages aligned with PGA: population initialization, fitness function, tournament selection, crossover operators, mutation operators, subgroup exchange, and termination criteria. The proposed PGA-based HPO method is implemented for Long Short-Term Memory Neural Networks models and evaluated on two real-world Internet of Things (IoT) sensor time-series datasets. Comparative results demonstrate that the proposed method outperforms other mainstream HPO techniques in terms of time efficiency and prediction accuracy across different datasets. Another hyper-parameters tuning of LSTM also can be found in [35,36] which applied for time series prediction.

Above all the presented prediction models, be it single or hybrid models, this area is still wide open for improvement, particularly for such important data such as Earth surface temperature. In this study, the DL will be hybridized with Barnacles Mating Optimizer (BMO-DL). The goal of the hybridization is to address the limitation of DL, where, despite the impressive success of DL techniques, the performance of these models heavily depends on the optimization of their internal parameters, namely weights and biases. Any inappropriate values set to the parameters will directly affect the overall performance of the DL model. Therefore, here, the BMO will be served as an optimization tool for the DL parameters. BMO is chosen due to its superiority in addressing various optimization issues which includes in telecommunication networks [37], information security [38], power system [39,40], finance [41], information retrieval [42], and many more.

The key contribution of the proposed method is as follow:

i. Through the integration of DL and the optimization method, BMO, the proposed approach effectively resolves challenges associated with hyperparameter optimization in conventional methods and mitigates the risk of overfitting minor differences in deep learning approaches.
ii. The proposed BMO-DL surpasses recognized hybrid methods, as evidenced by experimental results on specified datasets, which demonstrate markedly lower error rates in comparison to DL optimized by Particle Swarm Optimization (PSO-DL), Harmony Search Algorithm (HSA-DL) and Ant Colony Optimization (ACO-DL).

The subsequent sections of this paper are organized as follows: In Section 2. A concise the development of BMO mathematical model is presented, followed by a description on Deep Learning in Section 3. Section 4 outlines the methodology adopted, encompassing data collection, training and testing, the hybrid BMO-DL model and evaluation. The acquired results are examined in Sections 5, and 6 offers the concluding remarks.

2. Barnacles Mating Optimizer

The BMO [43] is developed based on Evolutionary Algorithm (EA). Just like other EA based algorithms, it incorporates three phases: initialization, selection, and reproduction.

2.1. Initialization

During initialization phase, a population of potential solutions (barnacles) is created. The population vector can be represented in the following manner:

$$x = \begin{bmatrix} x_1^1 & \dots & x_1^N \\ \dots & \dots & \dots \\ x_n^1 & \dots & x_n^N \end{bmatrix} \tag{1}$$

In the context of the problem at hand, N represents the count of control variables, while n denotes the population size or, equivalently, the number of barnacles. The control variables in (1) are constrained within the upper and lower bounds defined for the problem as follows:

$$ub = [ub_1, \ \dots, \ ub_i] \tag{2}$$

$$lb = [lb_1, \ \dots, \ lb_i] \tag{3}$$

Where ub and lb refer to the upper and lower bounds of the i th variable. The vector x is initially evaluated, and subsequently, a sorting process is conducted to identify the best solution found thus far, positioning it at the top of the vector x.

2.2. Selection process

The BMO adopts a distinctive mating selection approach in contrast to traditional evolutionary algorithms like Genetic Algorithm (GA) [44] and Differential Evolution (DE) [45]. In this approach, the selection of two barnacles is based on the length of their penises, pl. The selection process mimics the behavior of barnacles which are based on the following assumptions:

(i) The selection process is carried out randomly, but it is constrainedly by the pl.
(ii) Each barnacles can both contribute and receive sperm from other barnacles, with the restriction that each barnacles can be fertilized by only one other barnacle at a given time, despite the

possibility of multiple fertilizations by different males in real-life situations [46].

(iii) In the event that the selection process happens to choose the same barnacle, implying the potential for self-mating or self-fertilization, it's worth noting that, based on [47], self-mating is an infrequent occurrence, even though barnacles possess both male and female reproductive capabilities. Therefore, this paper does not consider self-mating, and at this point, no new offspring will be generated.

(iv) If the selection in a specific iteration exceeds the designated *pl*, the sperm-casting process is initiated.

It is worth mentioning that based on the aforementioned assumptions, the BMO algorithm inherently incorporates both exploitation (in points no. 1 and 2) and exploration (in point no. 4) processes.

The process of selecting ten barnacles for mating can be depicted in Fig. 1. The figure illustrates that the current best solution is positioned at the top among the candidate solutions in vector X. Let's assume that the maximum penis length of barnacles is seven time greater than their size (pl=7). Therefore, in a given iteration, barnacle #1 can only mate with one of the barnacles #2-#7. If barnacle #1 selects barnacle #8, it exceeds the limit, and the standard mating process does not occur. Consequently, the offspring generation proceeds through the sperm-casting process (exploration), which will be explained in more detail later. It'; s important to note that this operation is based on virtual distances and is not directly related to the real physical distances between barnacles. The following straightforward selection criteria are expressed in mathematical notation:

$$barnacle_d = randperm(n) \tag{4}$$

$$barnacle_m = randperm(n) \tag{5}$$

In this context, *barnacle_d* and *barnacle_m* represent the parent barnacles to be mated, with n denoting the population size.

2.3. Reproduction

The reproductive mechanism in BMO differs slightly from conventional EA. Given the absence of precise equations or formulas for barnacle reproduction, BMO primarily focuses on the inheritance traits and genotype frequencies of the parent barnacles to generate offspring, drawing inspiration from the Hardy-Weinberg principle. To illustrate the straightforward nature of the proposed BMO, the following expressions are put forth for the creation of new offspring variables derived from the genetic makeup of barnacle parents:

$$x_i^{N\_new} = p x_{barnacle_d}^N + q x_{barnacle\_m}^N \tag{6}$$

In this context, p represents pseudo-random numbers following a normal distribution within the range of [0, 1]. On the other hand, q=(1-p), $x_{barnacle_d}^N$ and $x_{barnacle_m}^N$correspond to the variables of the father (Dad) and mother (Mum) barnacles, respectively, which have been selected in (4) and (5). Essentially, p and q serve as indicators of the proportion of characteristics inherited from Dad and Mum that are incorporated into the generation of new offspring. In this manner, the offspring's traits are determined by the probabilities defined by the random numbers ranging from 0 to 1. For illustrative purposes, assuming p is 0.6 (randomly

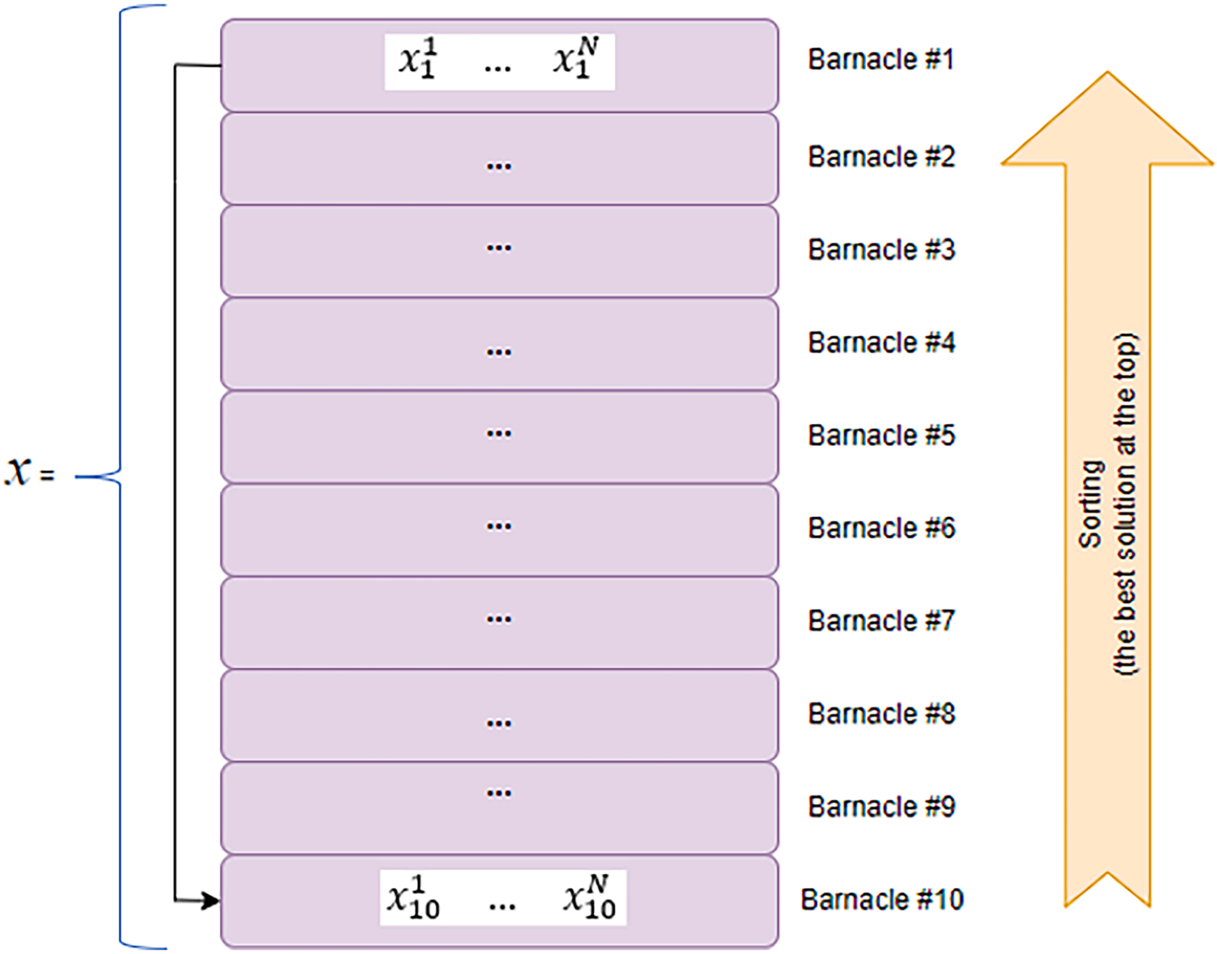


Fig. 1. Selection of mating process in BMO [43].

generated), this indicates that 60 % of Dad's traits and 40 % of Mum's traits are incorporated into the new offspring generation.

It's important to emphasize that the value of *pl* (penis length) plays a pivotal role in determining both exploitation and exploration processes within the algorithm. When the selection of barnacles for mating falls within the range of the father barnacle's penis length (as specified in (6)), it triggers the exploitation process, as mentioned in the selection phase. As detailed earlier, the sperm-casting mechanism is considered the exploration process in BMO. The sperm-casting process comes into play when the selection of barnacles for mating exceeds the initially set *pl* value (refer to Section 2.2). The description of the sperm-casting process is presented as follows:

$$x_i^{n\_new} = rand() \; \times x_{barnacle\_m}^{N} \tag{7}$$

Where, rand() is the random number between [0,1]. Fig. 2 shows the BMO pseudo-code.

3. Deep learning

Deep learning (DL) models excel in capturing complex nonlinear relationships, showcasing outstanding predictive accuracy, and proving especially beneficial for analyzing large building energy datasets. They adeptly capture long-term dependencies and temporal dynamics, resulting in enhanced prediction performance for time-series-related forecasts. Prominent DL models encompass Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), Long Short-Term Memory Neural Networks (LSTM), and Gate Recurrent Units (GRU) [48].

To predict Earth's surface temperature, a DL model, namely fixed forward neural networks (FFNN) is applied in this study. The DL architecture consists of an input layer, two hidden layers with 7 hidden neurons each, and an output layer. Details regarding the input and output are provided in Section 4.1. Given the significant dependence of DL on specific values of weights and biases, this research shifts away from using the Back Propagation (BP) algorithm for network training. Instead, the study adopts the BMO approach, as detailed in Sections 2 and 4.3.

4. Methodology

This section outlines the application of BMO-DL for predicting Earth surface temperature. It encompasses data collection, data normalization, the optimization of network weights using BMO, benchmarking the prediction model, and the subsequent evaluation. Fig. 3 illustrates the model of the proposed prediction model based on BMO-DL, which was implemented for the time series prediction of global surface temperature.

4.1. Dataset description

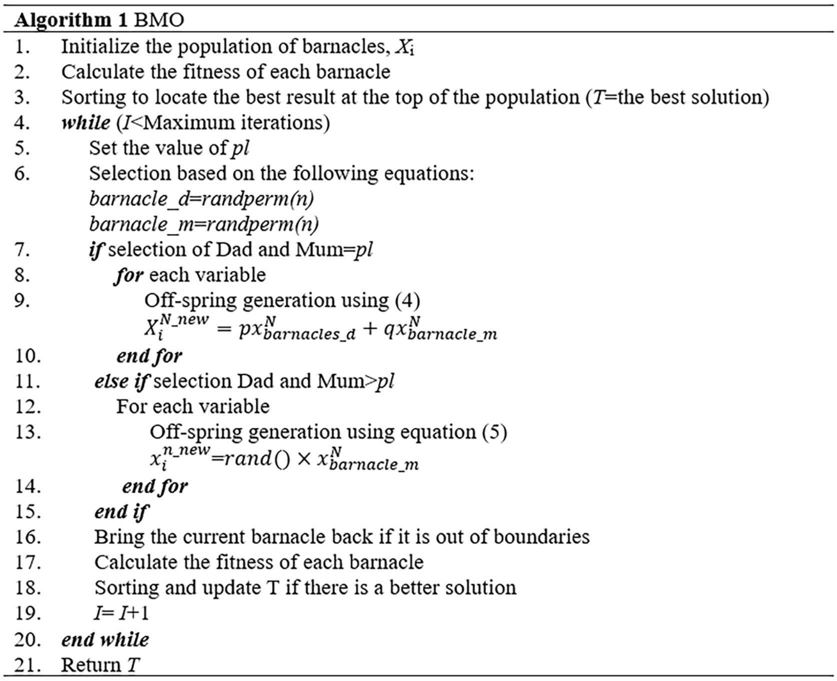
This dataset is gathered on a monthly basis, spanning from in January 1750 to December 2015, comprising 3192 data rows and 0 attributes variables. For the experiment conducted, 1992 data rows from January 1850 to December 2015 were utilized for time-series prediction. The target variable selected was the average temperature of land and ocean surfaces (Earth surface temperature). The date attribute and the last column of attribute variables were excluded, and the remaining seven attributes were employed as independent variables. The selected 1992 data rows were divided into two parts, with the initial 80 % used as the training set, and the remaining 20 % allocated as the test set. The time-series dataset is publicly accessible in [49]. Table 1 shows the sample of raw dataset:

4.2. Data normalization

Prior to inputting the datasets into the prediction algorithm, normalization was applied using Min-Max Normalization. The objective was to prevent smaller values from being overshadowed by larger input values.

$$MinMax = (v - min_a)/(max_a - min_a) \tag{8}$$

Where *v* is the respective value of the attribute, while min_a and max are the minimum and maximum of the given attribute, respectively. Data tabulated in Table 2 shows the normalized values of Table 1.



```
Algorithm 1 BMO
Initialize the population of barnacles, $X_i$
Calculate the fitness of each barnacle
Sorting to locate the best result at the top of the population ($T$=the best solution)
while ($I$<Maximum iterations)
    Set the value of $pl$
    Selection based on the following equations:
    $barnacle\_d=randperm(n)$
    $barnacle\_m=randperm(n)$
    if selection of Dad and Mum=$pl$
        for each variable
            Off-spring generation using (4)
            $X_i^{N\_new} = px_{barnacles\_d}^{N} + qx_{barnacle\_m}^{N}$
        end for
    else if selection Dad and Mum>$pl$
        For each variable
            Off-spring generation using equation (5)
            $x_i^{n\_new}=rand() \times x_{barnacle\_m}^{N}$
        end for
    end if
    Bring the current barnacle back if it is out of boundaries
    Calculate the fitness of each barnacle
    Sorting and update T if there is a better solution
    $I= I+1$
end while
Return $T$
```

Fig. 2. Pseudo code of BMO.

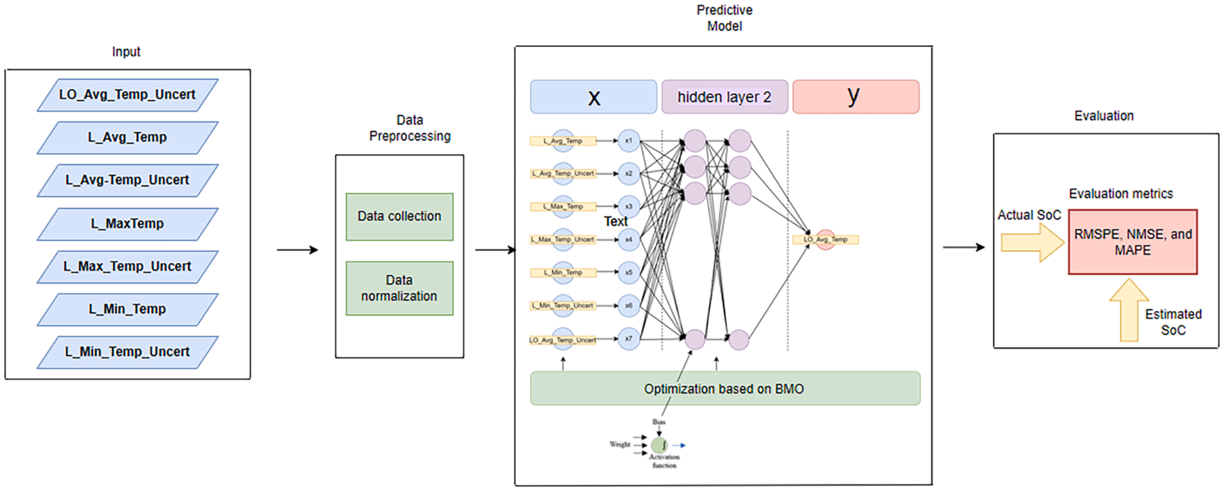


Fig. 3. Earth surface temperature using BMO-DL.

Table 1
Sample of raw datasets.

LO_Avg_Temp_Uncert	L_Avg_Temp	L_Avg-Temp_Uncert	L_MaxTemp	L_Max_Temp_Uncert	L_Min_Temp	L_Min_Temp_Uncert	L_O_Avg_Temp
0.367	0.749	1.105	8.242	1.738	−3.206	2.822	12.833
0.414	3.071	1.275	9.97	3.007	−2.291	1.623	13.588
0.341	4.954	0.955	10.347	2.401	−1.905	1.41	14.043
0.267	7.217	0.665	12.934	1.004	1.018	1.329	14.667
0.249	10.004	0.617	15.655	2.406	3.811	1.347	15.507
0.245	13.15	0.614	18.946	2.817	7.106	0.857	16.353
0.238	14.492	0.614	19.233	2.84	8.014	0.786	16.783
0.28	14.039	0.802	18.477	2.079	7.406	1.086	16.718
0.254	11.505	0.675	15.846	2.692	4.533	1.798	15.886
0.297	8.091	0.863	13.189	2.338	2.013	2.133	14.831

*LandAndOceanAverageTemperatureUncertainty=LO_Avg_Temp_Uncert, Land AverageTemperature= *L*_Avg_Temp, LandAverageTemperatureUncertainty = *L*_Avg_Temp_Uncert,
LandMaxTemprature = *L*_Max_Temp, LandMaxTemperatureUncertainty=*L*_Max_Temp_Uncert, LandMinTemperature = *L*_Min_Temp_Uncert,
LandAndOceanAverageTemperature = *L*_O_Avg_Temp.

4.3. Optimization of network weights using BMO algorithm

For the task of prediction, a Deep learning (DL) model is applied. This DL model is a feedforward (FFNN), supervised learning network, comprising an input layer, two hidden layers each with 5 hidden neurons, and an output layer. The input layer incorporates variables such as *LO_Avg_Temp_Uncert*, *L_Avg_Temp*, *L_Avg_Temp_Uncert*, *L_Max_Temp*, *L_Max_Temp_Uncert*, and *L_Min_Temp_Uncert*, while the output is *L_O_Avg_Temp*, as described in Sections 4.1 and 4.2. Given the significant impact of specific weights (initial weights of the neural network) and biases (the biases associated with each neuron in the network) on DL outcomes, this study diverges from employing the Back Propagation (BP) algorithm for network training. Instead, it opts for the BMO approach, as detailed in Section 2. The initial settings of weights and biases had an impact on the model's convergence speed and final accuracy. Proper initialization led to more stable and faster convergence. By using BMO to optimize these specific hyperparameters, the model achieved enhanced performance in terms of accuracy, convergence speed, and generalization. The sensitivity analysis highlighted the importance of each hyperparameter and demonstrated how BMO

Table 2
Sample of normalized datasets.

LO_Avg_Temp_Uncert	L_Avg_Temp	L_Avg-Temp_Uncert	L_MaxTemp	L_Max_Temp_Uncert	L_Min_Temp	L_Min_Temp_Uncert	L_O_Avg_Temp
0.7831	0.0229	0.7346	0.1519	0.3913	0.1455	0.8042	0.0697
0.8964	0.1769	0.8512	0.2639	0.6845	0.2061	0.457	0.2167
0.7205	0.3018	0.6317	0.2884	0.5445	0.2316	0.3953	0.3053
0.5422	0.4519	0.4328	0.4562	0.2218	0.4249	0.3719	0.4268
0.4988	0.6367	0.3999	0.6326	0.5456	0.6096	0.3771	0.5903
0.4892	0.8453	0.3978	0.846	0.6406	0.8275	0.2352	0.7551
0.4723	0.9343	0.3978	0.8647	0.6459	0.8875	0.2146	0.8388
0.5735	0.9043	0.5267	0.8156	0.4701	0.8473	0.3015	0.8261
0.5108	0.7362	0.4396	0.645	0.6117	0.6573	0.5077	0.6641
0.6145	0.0229	0.7346	0.1519	0.3913	0.1455	0.8042	0.0697

*LandAndOceanAverageTemperatureUncertainty=LO_Avg_Temp_Uncert, Land AverageTemperature= *L*_Avg_Temp, LandAverageTemperatureUncertainty = *L*_Avg_Temp_Uncert,
LandMaxTemprature = *L*_Max_Temp, LandMaxTemperatureUncertainty=*L*_Max_Temp_Uncert, LandMinTemperature = *L*_Min_Temp_Uncert,
LandAndOceanAverageTemperature = *L*_O_Avg_Temp.

effectively balanced them to improve the forecasting capability of the deep learning model.

The architecture of the DL model for predicting Earth surface temperature is visually depicted in Fig. 4.

The determination of the number of iterations was based on an iterative trial-and-error approach. In this study, the number of iterations was established as 1000. This decision was informed by observations indicating that beyond this threshold, there was no significant improvement observed through BMO-DL.

Regarding the optimization of network weights and biases, the search space was defined within the [-1, 1] range during finding the optimal solution. Subsequently, the optimizer systematically explored this space to identify the combination of weights and biases that would optimize the neural network's performance relative to a specified objective function. This methodology also implicitly includes a constraint within the study.

4.4. Benchmark prediction model

To assess the efficacy of the proposed BMO-DL, two hybrid models were selected namely DL optimized by Particle Swarm optimization [50] (PSO-DL), Harmony Search Algorithm [51] (HSA-DL) and Ant Colony Optimization (ACO-DL). Besides, a statistical model was selected too, namely Autoregressive Integrated Moving Average (ARIMA). PSO is based on the concept of swarm, where a population of a potential solutions (called particles) moves through the search space to find the optimal solution. Each particle represents a potential solution to the optimization problem. On the other hand, the HSA is a nature inspired optimization algorithm that draws its inspiration from the process of musicians harmonizing their instruments to find the best melody. ACO, inspired by the foraging behavior of ants, is a population-based optimization technique that uses pheromone trails to find optimal paths through graphs. In ACO, artificial ants build solutions by traversing paths and depositing pheromones, which guide subsequent ants towards promising solutions, effectively finding optimal or near-optimal solutions to complex problems. Meanwhile, ARIMA is a widely used

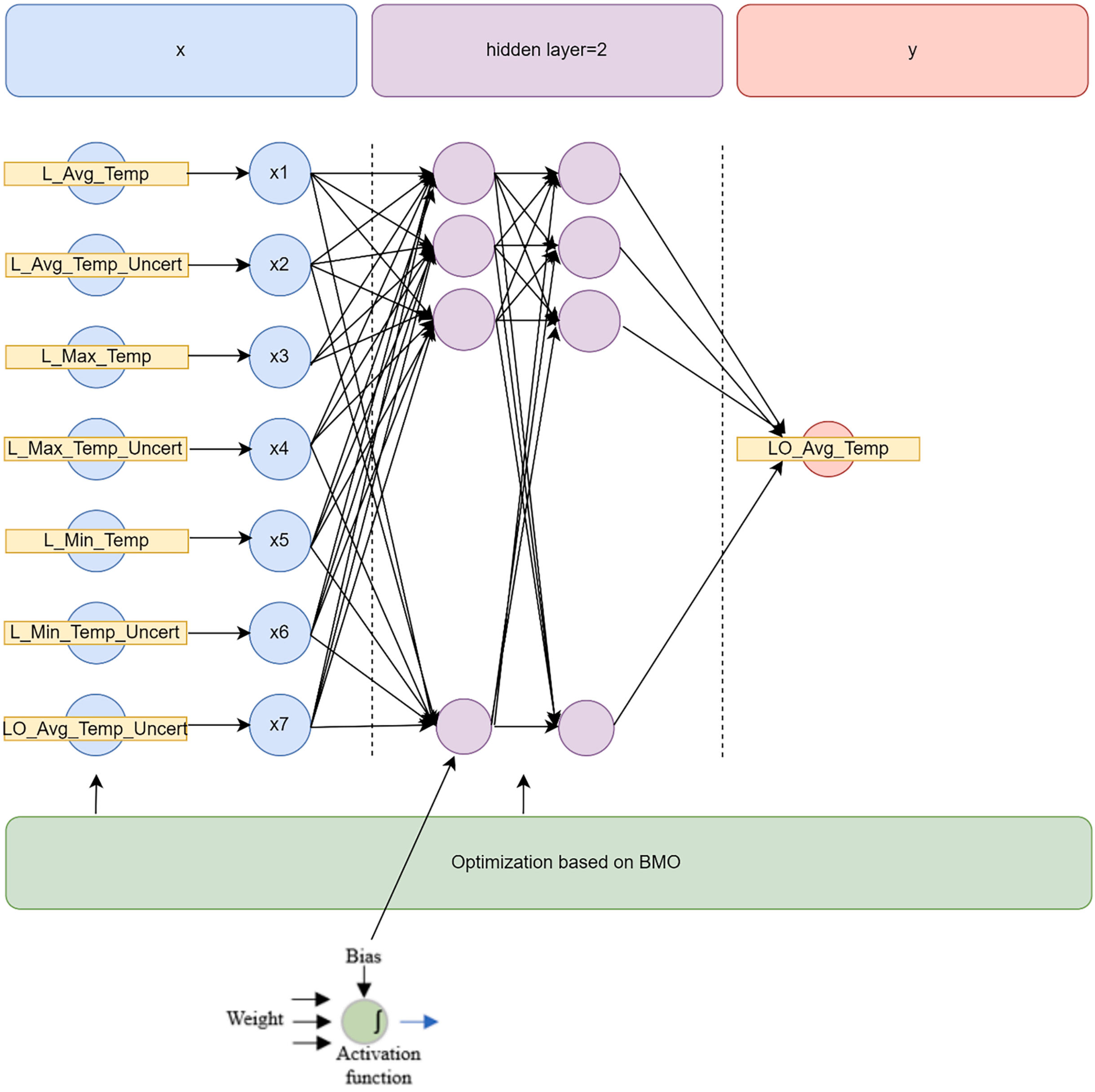


Fig. 4. BMO-DL.

forecasting technique that models time series data by accounting for its autocorrelations and differencing to make it stationary.

4.5. Model performance evaluation metrics

In this study, the Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and coefficient of determination (R^2) were applied as metrics to evaluate the predictive performance of the proposed method on the respective dataset. The calculation formulas of these metrics are presented below:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widetilde{y}_i)^2 \tag{9}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widetilde{y}_i)^2} \tag{10}$$

$$R^2 = 1\text{-}\frac{\sum_{1=1}^{N}(y_i - \widetilde{y}_i)^2}{\sum_{1=1}^{N}(y_i - \overline{y})^2} \tag{11}$$

Here, y_i denotes the true value of the Earth surface temperature, $\widetilde{y}_i$ signifies the ultimate predicted value within the dataset, $\widetilde{y}_i$ is the mean of observed data, and N stands for the total number of samples in the test set. It is widely accepted that smaller values for MAE and RMSE performance evaluation indicators are associated with greater predictive accuracy of the model, while for R^2, a value to 1 indicates that the regression predictions better fit the data.

5. Results and discussion

The experimental evaluation of the proposed BMO-DL approach was carried out on a computational system running Windows 11 operating system, equipped with an AMD Ryzen 5 5500H processor, 16 GB of RAM, and integrated Radeon Graphics.

Determining the optimal number of hidden neurons is crucial for achieving accurate prediction values in DL. To address this, the present study conducted experiments with three different hidden neuron values: 5, 7, and 9. After completing the experiments, it was observed that the BMO-DL with 7 hidden neurons yielded the lowest MAE and RMSE. Notably, the maximum error value appears to align well with the errors generated.

Based on the information provided in Table 3, BMO-DL produced a denormalized Mean Absolute Error (MAE) of 0.2019 and a Root Mean Squared Error (RMSE) of 0.2374 when configured with 7 hidden neurons. Conversely, the utilization of 5 hidden neurons yielded the maximum MAE and RMSE, recording values of 0.2227 and 0.2640, respectively. The corresponding results for normalized values can be found in Table 4.

By employing the configuration with 7 hidden neurons, the experiment proceeded to compare the outcomes with DL optimized by PSO (PSO-DL), HSA (HSA-DL), ACO-DL and ARIMA. The denormalized results obtained are organized in Table 5. As depicted in the table, BMO-DL exhibited superior performance compared to the four identified algorithms, as all other methods yielded higher error rates in terms of MAE and RMSE, with ARIMA ranking the lowest. The outcomes for normalized values are presented in Table 6.

Table 3
Comparison of different hidden neurons configuration using BMO-DL for denormalized data.

	5-hidden neurons	7-hidden neurons	9-hidden neurons
MAE	0.2227	**0.2019**	0.2024
RMSE	0.2640	**0.2374**	0.2369
Maximum Error	0.6367	**0.5308**	0.5690
R^2	0.9540	0.9628	**0.9629**

Table 4
Comparison of different hidden neurons configuration using BMO-DL for normalized data.

	5-hidden neurons	7-hidden neurons	9-hidden neurons
MAE	0.0434	**0.0393**	0.0394
RMSE	0.0514	**0.0462**	0.0461
Maximum Error	0.1240	**0.1033**	0.1108
R^2	0.9540	0.9628	**0.9629**

Table 5
BMO-DL vs. PSO-DL vs. HSA-DL using denormalized data.

	BMO-DL	PSO-DL	HSA-DL	ARIMA	ACO-DL
MAE	**0.2019**	0.2629	0.2828	0.5248	0.2432
RMSE	**0.2374**	0.2972	0.3246	0.6073	0.2765
Maximum Error	**0.5308**	0.6402	0.7167	1.3069	0.5946
R^2	**0.9628**	0.9417	0.9304	0.7564	0.9495

The predictions generated by BMO-DL seem to closely match the target values, resulting in relatively small residuals. As illustrated in Table 7, predictions from all methods are presented covering days 1638 to 1651, constituting 3.5 % of the entire testing phase. During this period, an exception arises on days 1643 and 1644, where BMO-DL shows notably larger residuals of 0.1118 and 0.0929, respectively, indicating a more substantial prediction error for those specific days. Meanwhile, for PSO-DL, HSA-DL, ACO-DL and ARIMA consistently yield significantly larger residuals compared to BMO-DL throughout most of the testing phase.

Besides the obtained results in MAE and RMSE, the R^2 results were also recorded, where for both normalized and denormalized data, the higher R^2 value of BMO-DL is consistent with its lower MAE and RMSE values compared to the other models, further confirming its better performance. BMO-DL has the highest R^2 of 0.9628, indicating that this model has the best fit to the data compared to PSO-DL, HSA-DL, ACO-DL and ARIMA.

In general, a higher R^2 value that closer to 1 indicates a better fit of the model to the data, which is typically associated with lower MAE and RMSE values (better accuracy). The results align with this expectation, as BMO-DL, with the highest R^2, also has the lowest MAE and RMSE for both denormalized and normalized data.

The BMO-DL model exhibits robust performance on unseen data, with accuracy rates comparable to those achieved during training and testing phases. Evaluation metrics, including MAE, RMSE, and R^2, consistently demonstrate strong performance on previously unseen samples. Moreover, the model demonstrates promising generalization capabilities beyond the dataset used for training and testing. It effectively captures underlying patterns and relationships in diverse data distributions, as evidenced by extensive experimentation and validation. These findings suggest the model's potential applicability to real-world scenarios, highlighting its versatility and effectiveness across varied contexts.

Fig. 5 depicts the comparison between BMO-DL, PSO-DL, HSA-DL, ARIMA and ACO-DL throughout the entire testing phase. The illustration highlights that BMO-DL, indicated by a cross mark, consistently generated prediction values that closely aligned with the actual values, particularly in the period from day 1600 to 1650.

Table 8 presents the results of *t*-tests comparing different methods in

Table 6
BMO-Dl vs. PSO-DL vs. HSA-DL using normalized data.

	BMO-DL	PSO-DL	HSA-DL	ARIMA	ACO-DL
MAE	**0.0393**	0.0512	0.0551	0.1022	0.0473
RMSE	**0.0462**	0.0579	0.0632	0.1183	0.0538
Maximum Error	**0.1033**	0.1247	0.1395	0.2544	0.1157
R^2	**0.9628**	0.9417	0.9304	0.7564	0.9495

Table 7
Prediction vs. target values, and residuals of BMO-DL, PSO-DL, HAS-DL, ACO-DL and ARIMA.

							Residual				
Days	Target	BMO-DL	PSO-DL	HSA-DL	ARIMA	ACO-DL	BMO-DL	PSO-DL	HSA-DL	ARIMA	ACO-DL
1638	16.696	16.7311	16.5949	16.4110	16.7732	16.6063	**0.0351**	0.1011	0.2850	0.0772	0.0897
1639	16.991	16.9739	16.8059	16.8702	17.0205	16.8728	**0.0171**	0.1851	0.1208	0.0295	0.1182
1640	16.938	16.9082	16.7505	16.7268	16.8130	16.8081	**0.0298**	0.1875	0.2112	0.1250	0.1299
1641	16.285	16.2584	16.1807	15.9761	16.2075	16.1910	**0.0266**	0.1043	0.3089	0.0775	0.0940
1642	15.455	15.4382	15.4362	15.4748	15.3661	15.4988	**0.0168**	0.0188	0.0198	0.0889	0.0438
1643	14.407	14.5188	14.4452	14.5502	14.5117	14.4912	0.1118	**0.0382**	0.1432	0.1047	0.0842
1644	13.805	13.8979	13.8356	13.8835	13.8756	13.7555	0.0929	**0.0306**	0.0785	0.0706	0.0495
1645	13.758	13.6626	13.6413	13.6256	13.6270	13.5418	**0.0954**	0.1167	0.1324	0.1310	0.2162
1646	14.161	13.9751	13.9460	13.9320	13.8312	13.9223	**0.1859**	0.2150	0.2290	0.3298	0.2387
1647	14.538	14.2978	14.2953	14.2665	14.4343	14.3121	0.2402	0.2427	0.2715	**0.1037**	0.2259
1648	15.447	15.2947	15.2911	15.2516	15.2748	15.3374	0.1523	0.1559	0.1954	0.1722	**0.1096**
1649	16.252	16.1656	16.0773	15.9484	16.1252	16.1147	**0.0864**	0.1747	0.3036	0.1268	0.1373
1650	16.9	16.8582	16.7101	16.6305	16.7593	16.7454	**0.0418**	0.1899	0.2695	0.1407	0.1546
1651	17.296	17.1548	16.9716	17.0028	17.0065	17.0128	**0.1412**	0.3244	0.2932	0.2895	0.2832

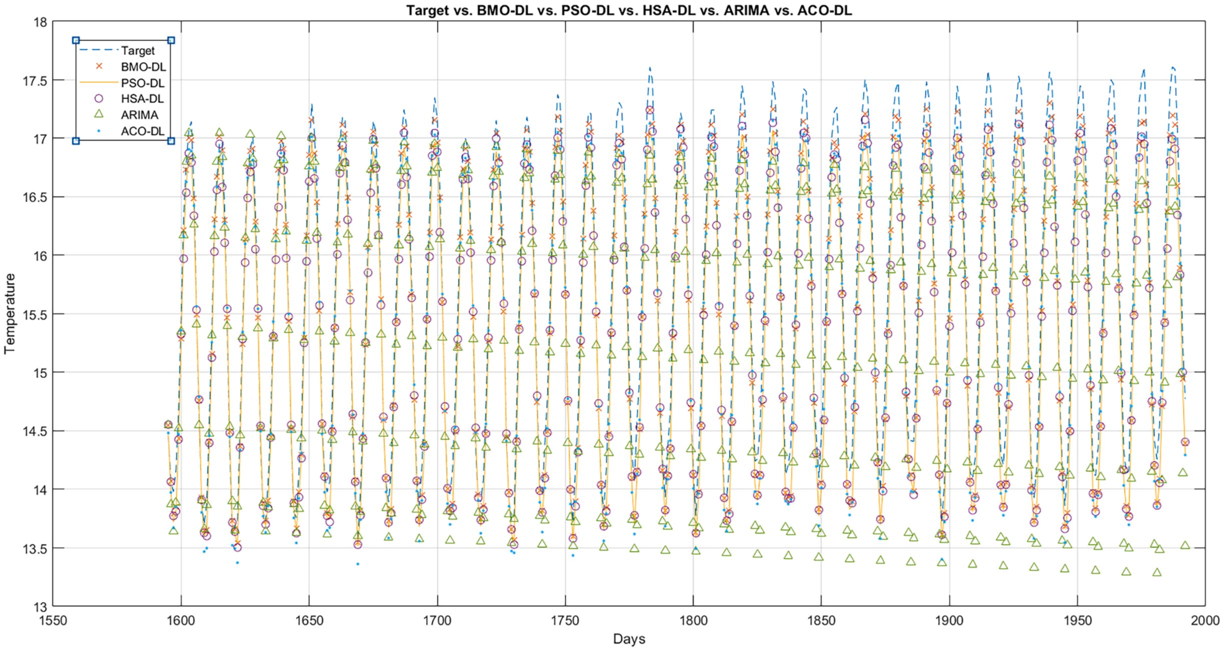


Fig. 5. Performance of BMO-DL compared to target and identified algorithms for testing phase.

terms of the probability (P) that the observed t-statistic is less than or equal to the calculated t-value in a two-tailed test. The methods compared are BMO-DL against four other methods: PSO-DL, HSA-DL, ACO-DL and ARIMA. The obtained *p*-values for both comparisons are extremely small, specifically 0.0000, indicating a statistically significant difference between the performance of BMO-DL and each of the other methods.

In terms of computational time, for the dataset comprising 3192 data points with seven input features and one output, the BMO-DL approach required a computational time of 32.9171 s for 1000 iterations. While this computational time may seem reasonable for the given dataset size, it is important to note that the complexity of the BMO optimization process scales with the number of data points, features, and iterations required for convergence. Therefore, for larger datasets or more complex problems, the computational requirements may increase significantly. However, potential optimizations such as parallelization, GPU acceleration, or distributed computing could be explored to improve the computational efficiency of the BMO-DL approach, making it more feasible for deployment in real-world scenarios with stringent computational constraints.

Table 8
Paired sample t-Test.

Methods	P (*T*<=*t*) two-tail
BMO-DL vs. PSO-DL	0.0000
BMO-DL vs. HAS-DL	0.0000
BMO-DL vs. ARIMA	0.0000
BMO-DL vs. ACO-DL	0.0000

6. Conclusion

In this research, the BMO in combination with DL (BMO-DL) is employed as a predictive model for Earth's surface temperature. For experimental purposes, this study incorporated seven input variables that exerted an influence on the Earth's surface temperature. An exploration into the optimal number of hidden neurons is conducted through a series of experiments, encompassing configurations with 5, 7, and 9 hidden neurons. Subsequently, the effectiveness of the proposed BMO-DL is assessed by comparing it with two analogous hybrid methods, namely PSO-DL and HSA-DL. The evaluation is based on two performance indices, specifically MAE and RMSE. The results obtained demonstrate the superiority of BMO-DL, which exhibits significantly lower error rates. This superiority is substantiated by the outcomes of a *t*-test conducted for validation purposes. The findings highlight that the BMO is able to effectively optimize DL models, resulting in superior predictive accuracy, lower prediction errors, and a better fit to the data compared to the PSO and HSA algorithms used for the same task. This underscores the capability of BMO-DL as a proficient method for accurately predicting Earth's surface temperature, offering substantial benefits to the concerned parties.

Despite its promising performance, is not without limitations. Like other metaheuristic optimization techniques, BMO may suffer from premature convergence or getting trapped in local optima, especially when dealing with high-dimensional search spaces or complex problem landscapes. Additionally, the performance of BMO can be sensitive to its parameter settings, and finding the optimal parameter values may require additional tuning efforts, which can be time-consuming and computationally expensive. Furthermore, BMO, being a population-based algorithm, can be computationally demanding for problems with large search spaces or high-dimensional solutions, as it requires evaluating multiple candidate solutions simultaneously. These limitations will be considered when applying BMO to real-world optimization problems and interpreting the results obtained from the algorithm.

The dataset used for training and testing the proposed approach may also have inherent biases or limitations that could impact the generalizability of the results. The dataset may not adequately represent certain geographic regions, terrain types, or environmental conditions, leading to biased or skewed predictions for underrepresented areas or conditions. Additionally, the dataset itself may contain noise, outliers, or missing values, which could influence the training process and the accuracy of the predictions made by the model. Furthermore, the dataset may be skewed or imbalanced, with an overrepresentation or underrepresentation of certain classes or regions, potentially leading to biased predictions for the underrepresented classes or areas.

In conclusion, the hybrid BMO-DL approach represents a notable advancement, demonstrated by its improved performance compared to established hybrid algorithms. As research continues to evolve, this study lays a foundation for future endeavors, promoting ongoing innovation and progress in the relevant domain.

Data availability declaration

Data sharing not applicable to this article as datasets can be obtained from the literature.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to express our gratitude to Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Grant No.: RDU220379, for funding this research.

References

[1] S. Kumar, A novel hybrid machine learning model for prediction of CO_2 using socio-economic and energy attributes for climate change monitoring and mitigation policies, Ecol. Inform. 77 (2023) 102253, https://doi.org/10.1016/j.ecoinf.2023.102253, 2023/11/01/.

[2] G. Roshan, A. Ghanghermeh, V. Mohammadnejad, P. Fdez-Arróyabe, A. Santurtún, Predicting climate change impact on hospitalizations of cardiovascular patients in Tabriz, Urban Clim. 44 (2022) 101184, https://doi.org/10.1016/j.uclim.2022.101184, 2022/07/01/.

[3] A. Singh, A. Haghverdi, Development and evaluation of temperature-based deep learning models to estimate reference evapotranspiration, Artif. Intell. Agric. 9 (2023) 61–75, https://doi.org/10.1016/j.aiia.2023.08.003, 2023/09/01/.

[4] S. Chen, et al., Occupant-centric dynamic heating and cooling loads simplified prediction model for urban community at energy planning stage, Sustain. Cities Soc. 90 (2023) 104406, https://doi.org/10.1016/j.scs.2023.104406, 2023/03/01/.

[5] M. Koc, A. Acar, Investigation of urban climates and built environment relations by using machine learning, Urban Clim. 37 (2021) 100820, https://doi.org/10.1016/j.uclim.2021.100820, 2021/05/01/.

[6] J. Siqi, W. Yuhong, C. Ling, B. Xiaowen, A novel approach to estimating urban land surface temperature by the combination of geographically weighted regression and deep neural network models, Urban Clim. 47 (2023) 101390, https://doi.org/10.1016/j.uclim.2022.101390, 2023/01/01/.

[7] X. Zhou, J. Wang, Y. Liu, Q. Duan, Deep learning with PID residual elimination network for time-series prediction of water quality in aquaculture industry, Comput. Electron. Agric. 212 (2023) 108125, https://doi.org/10.1016/j.compag.2023.108125, 2023/09/01/.

[8] G. Liu, K. Zhong, H. Li, T. Chen, Y. Wang, A state of art review on time series forecasting with machine learning for environmental parameters in agricultural greenhouses, Inf. Process. Agric. (2022), https://doi.org/10.1016/j.inpa.2022.10.005, 2022/10/28/.

[9] Q.C. Li, S.W. Xu, J.Y. Zhuang, J.J. Liu, Y. Zhou, Z.X. Zhang, Ensemble learning prediction of soybean yields in China based on meteorological data, J. Integr. Agric. 22 (6) (2023) 1909–1927, https://doi.org/10.1016/j.jia.2023.02.011, 2023/06/01/.

[10] S.C.A. Houetohossou, V.R. Houndji, C.G. Hounmenou, R. Sikirou, R.L.G. Kakaï, Deep learning methods for biotic and abiotic stresses detection and classification in fruits and vegetables: state of the art and perspectives, Artif. Intell. Agric. 9 (2023) 46–60, https://doi.org/10.1016/j.aiia.2023.08.001, 2023/09/01/.

[11] M.T. Ahad, Y. Li, B. Song, T. Bhuiyan, Comparison of CNN-based deep learning architectures for rice diseases classification, Artif. Intell. Agric. 9 (2023) 22–35, https://doi.org/10.1016/j.aiia.2023.07.001, 2023/09/01/.

[12] P.I. Ritharson, K. Raimond, X.A. Mary, J.E. Robert, A. J, DeepRice: a deep learning and deep feature based classification of Rice leaf disease subtypes, Artif. Intell. Agric. 11 (2024) 34–49, https://doi.org/10.1016/j.aiia.2023.11.001, 2024/03/01/.

[13] S. MirhoseiniNejad, G. Badawy, D.G. Down, Holistic thermal-aware workload management and infrastructure control for heterogeneous data centers using machine learning, Future Gener. Comput. Syst. 118 (2021) 208–218, https://doi.org/10.1016/j.future.2021.01.007, 2021/05/01/.

[14] B.A. Alkhaleel, Machine learning applications in the resilience of interdependent critical infrastructure systems—A systematic literature review, Int. J. Crit. Infrastruct. Prot. 44 (2024) 100646, https://doi.org/10.1016/j.ijcip.2023.100646, 2024/03/01/.

[15] M.H. Sulaiman, Z. Mustaffa, N.F. Zakaria, M.M. Saari, Using the evolutionary mating algorithm for optimizing deep learning parameters for battery state of charge estimation of electric vehicle, Energy 279 (2023) 128094, https://doi.org/10.1016/j.energy.2023.128094, 2023/09/15/.

[16] E. Gulay, M. Sen, O.B. Akgun, Forecasting electricity production from various energy sources in Türkiye: a predictive analysis of time series, deep learning, and hybrid models, Energy 286 (2024) 129566, https://doi.org/10.1016/j.energy.2023.129566, 2024/01/01/.

[17] C. van Zyl, X. Ye, R. Naidoo, Harnessing eXplainable artificial intelligence for feature selection in time series energy forecasting: a comparative analysis of Grad-CAM and SHAP, Appl. Energy 353 (2024) 122079, https://doi.org/10.1016/j.apenergy.2023.122079, 2024/01/01/.

[18] S. Çakır, Renewable energy generation forecasting in Turkey via intuitionistic fuzzy time series approach, Renew. Energy 214 (2023) 194–200, https://doi.org/10.1016/j.renene.2023.05.132, 2023/09/01/.

[19] Y. Zhang, S. Ragettli, P. Molnar, O. Fink, N. Peleg, Generalization of an encoder-decoder LSTM model for flood prediction in ungauged catchments, J. Hydrol. 614 (2022) 128577, https://doi.org/10.1016/j.jhydrol.2022.128577, 2022/11/01/.

[20] S. Bhardwaj, E. Chandrasekhar, P. Padiyar, V.M. Gadre, A comparative study of wavelet-based ANN and classical techniques for geophysical time-series forecasting, Comput. Geosci. 138 (2020) 104461, https://doi.org/10.1016/j.cageo.2020.104461, 2020/05/01/.

[21] Ü.Ç. Büyükşahin, Ş. Ertekin, Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition, Neurocomputing 361 (2019) 151–163, https://doi.org/10.1016/j.neucom.2019.05.099, 2019/10/07/.

[22] V.N. Vapnik, The Nature of Statistical Learning Theory, 2nd ed., Springer-Verlag, New York, 1995.

[23] A.M. Gómez-Orellana, D. Guijo-Rubio, J. Pérez-Aracil, P.A. Gutiérrez, S. Salcedo-Sanz, C. Hervás-Martínez, One month in advance prediction of air temperature from Reanalysis data with explainable Artificial Intelligence techniques, Atmos. Res. 284 (2023) 106608, https://doi.org/10.1016/j.atmosres.2023.106608, 2023/03/15/.

[24] W. Zhang, et al., A deep learning method for real-time bias correction of wind field forecasts in the Western North Pacific, Atmos. Res. 284 (2023) 106586, https://doi.org/10.1016/j.atmosres.2022.106586, 2023/03/15/.

[25] D. Tang, Y. Zhan, F. Yang, A review of machine learning for modeling air quality: overlooked but important issues, Atmos. Res. 300 (2024) 107261, https://doi.org/10.1016/j.atmosres.2024.107261, 2024/04/15/.

[26] M.J. Jiménez-Navarro, M. Martínez-Ballesteros, F. Martínez-Álvarez, G. Asencio-Cortés, PHILNet: a novel efficient approach for time series forecasting using deep

learning, Inf. Sci. 632 (2023) 815–832, https://doi.org/10.1016/j.ins.2023.03.021, 2023/06/01/.

[27] X.y. Fu, et al., Enhanced machine learning model via twin support vector regression for streamflow time series forecasting of hydropower reservoir, Energy Rep. 10 (2023) 2623–2639, https://doi.org/10.1016/j.egyr.2023.09.071, 2023/11/01/.

[28] Z.K. Feng, W.J. Niu, X.Y. Wan, B. Xu, F.L. Zhu, J. Chen, Hydrological time series forecasting via signal decomposition and twin support vector machine using cooperation search algorithm for parameter identification, J. Hydrol. 612 (2022) 128213, https://doi.org/10.1016/j.jhydrol.2022.128213, 2022/09/01/.

[29] A. Saeed, A. Alsini, D. Amin, Water quality multivariate forecasting using deep learning in a West Australian estuary, Environ. Model. Softw. 171 (2024) 105884, https://doi.org/10.1016/j.envsoft.2023.105884, 2024/01/01/.

[30] F. Di Nunno, S. Zhu, M. Ptak, M. Sojka, F. Granata, A stacked machine learning model for multi-step ahead prediction of lake surface water temperature, Sci. Total Environ. 890 (2023) 164323, https://doi.org/10.1016/j.scitotenv.2023.164323, 2023/09/10/.

[31] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. d. Freitas, Taking the human out of the loop: a review of bayesian optimization, Proc. IEEE 104 (1) (2016) 148–175, https://doi.org/10.1109/JPROC.2015.2494218.

[32] Z. Mustaffa, M.H. Sulaiman, Enhancing battery state of charge estimation through hybrid integration of Barnacles Mating Optimizer with deep learning, Frankl. Open 5 (2023) 100053, https://doi.org/10.1016/j.fraope.2023.100053, 2023/12/01/.

[33] C. Erden, Genetic algorithm-based hyperparameter optimization of deep learning models for PM2.5 time-series prediction, Int. J. Environ. Sci. Technol. 20 (2023) 2959–2982.

[34] D. Wu, Q. Guan, Z. Fan, H. Deng, T. Wu, AutoML with parallel genetic algorithm for fast hyperparameters optimization in efficient IoT time series prediction, IEEE Trans. Ind. Inform. 19 (9) (2023) 9555–9564, https://doi.org/10.1109/TII.2022.3231419.

[35] N. Bakhashwain, A. Sagheer, Online tuning of hyperparameters in deep LSTM for time series applications, Int. J. Intell. Eng. Syst. (2021), https://doi.org/10.22266/ijies2021.0228.21.

[36] W. Zheng, H. Jun, Multivariate time series prediction based on temporal change information learning method, IEEE Trans. Neural Netw. Learn. Syst. 34 (10) (2023) 7034–7048, https://doi.org/10.1109/TNNLS.2021.3137178.

[37] Z. Mohammadnejad, H.M.R. Al-Khafaji, A.S. Mohammed, S.R. Alatba, Energy optimization for optimal location in 5G networks using improved Barnacles Mating Optimizer, Phys. Commun. 59 (2023) 102068, https://doi.org/10.1016/j.phycom.2023.102068, 2023/08/01/.

[38] K.Z. Zamli, et al., Exploiting an elitist Barnacles Mating Optimizer implementation for substitution box optimization, ICT Express 9 (4) (2023) 619–627, https://doi.org/10.1016/j.icte.2022.11.005, 2023/08/01/.

[39] A. Norouzi, H. Shayeghi, J. Olamaei, Multi-objective allocation of switching devices in distribution networks using the Modified Barnacles Mating Optimization algorithm, Energy Rep. 8 (2022) 12618–12627, https://doi.org/10.1016/j.egyr.2022.09.028, 2022/11/01/.

[40] M.H. Sulaiman, Z. Mustaffa, Solving optimal power flow problem with stochastic wind–solar–small hydro power using Barnacles Mating Optimizer, Control Eng. Pract. 106 (2021) 104672, https://doi.org/10.1016/j.conengprac.2020.104672, 2021/01/01/.

[41] Z. Mustaffa, M.H. Sulaiman, Stock price predictive analysis: an application of hybrid Barnacles Mating Optimizer with Artificial Neural Network, Int. J. Cogn. Comput. Eng. 4 (2023) 109–117, https://doi.org/10.1016/j.ijcce.2023.03.003, 2023/06/01/.

[42] N. Pughazendi, P.V. Rajaraman, M.H. Mohammed, Graph sample and aggregate attention network optimized with Barnacles Mating Algorithm based sentiment analysis for online product recommendation, Appl. Soft Comput. 145 (2023) 110532, https://doi.org/10.1016/j.asoc.2023.110532, 2023/09/01/.

[43] M.H. Sulaiman, Z. Mustaffa, M.M. Saari, H. Daniyal, Barnacles Mating Optimizer: a new bio-inspired algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 87 (2020) 103330, https://doi.org/10.1016/j.engappai.2019.103330, 2020/01/01/.

[44] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, John, Wiley & Sons, Inc, 2004.

[45] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, <i data-test="journal-title" style="margin: 0px; box-sizing: inherit;">, J. Glob. Optim. 11 (1997) 341–359.

[46] M. Barazandeh, C.S. Davis, C.J. Neufeld, W. Coltman, David, A.R. Palmer, Something Darwin didn't know about barnacles: spermcast mating in a common stalked species, Proc. R. Soc. B 280 (2013).

[47] Yusa, et al., Adaptive evolution of sexual systems in pedunculate barnacles, Proc. R. Soc. B 279 (2012) 959–966.

[48] H. Liu, Y. Liu, X. Guo, H. Wu, H. Wang, Y. Liu, An energy consumption prediction method for HVAC systems using energy storage based on time series shifting and deep learning, Energy Build. 298 (2023) 113508, https://doi.org/10.1016/j.enbuild.2023.113508, 2023/11/01/.

[49] "Climate Change: Earth Surface Temperature Data", https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data (Accessed 15 September 2023).

[50] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the ICNN'95 International Conference on Neural Networks 4, 1995, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968, 27 Nov.-1 Dec. 1995vol. 4.

[51] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.