Franklin Open 5 (2023) 100053

Contents lists available at ScienceDirect

Franklin Open

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

Enhancing battery state of charge estimation through hybrid integration of barnacles mating optimizer with deep learning

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

[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
Machine learning
Optimization
Prediction
State of charge estimation

ABSTRACT

The precise determination of battery state of charge (SoC) holds paramount significance and has garnered considerable attention across diverse sectors, including academia. Accurate knowledge of the SoC percentage offers numerous advantages, ranging from optimizing travel planning to enhancing the efficiency and reliability of electric vehicle operations through effective battery management systems. In response to the growing importance of SoC estimation, this study introduces a hybrid approach called the Barnacles Mating Optimizer with Deep Learning (BMO-DL) for SoC of Nissan Leaf batteries. The conventional methods for SoC estimation often suffer from limitations in accuracy and robustness, leading to suboptimal EV performance and battery management. In contrast, BMO-DL leverages the power of BMO algorithm to fine-tune the hyperparameters of DL, which is subsequently employed for the actual estimation. This synergistic combination enhances the accuracy and reliability of SoC estimation. The estimation model takes three inputs: voltage, current and conducted charge to generate a single output, the SoC percentage. he study's findings underscore the superiority of BMO-DL by revealing its capability to achieve significantly better results compared to the other benchmarking methods identified. Notably, BMO-DL exhibits significantly lower error rates when compared to competing algorithms, thereby reinforcing its potential to advance the efficiency and reliability of electric vehicle operations while addressing the critical challenge of SoC prediction.

1. Introduction

In order to attain carbon neutrality, enhance air quality in urban areas, and fulfill consumer demands, governments worldwide are actively advocating for the adoption of innovative energy-efficient electric vehicles (EVs). With rapid progress in the accumulation of decommissioned lithium-ion batteries, environmental and economic concerns have garnered significant interest, leading to the evolution of research in the adequate EV charging infrastructure [1] as well as recycling process of these batteries [2]. One of the active issues in battery management systems is the battery State of Charge (SoC) estimation problem. Accurately estimating the SoC in batteries holds paramount significance across a spectrum of applications, spanning from electric vehicles (EVs) to renewable energy systems. The SOC of a battery represents the remaining available energy and plays a critical role in optimizing battery utilization, prolonging battery life, and ensuring safe and reliable operation. Given the uncertain driving patterns and the repetitive acceleration and deceleration of a vehicle, the mattery may experience significantly changing load requirements [3]. Therefore, the development of precise and efficient SoC estimation techniques is a subject of ongoing research in the field of energy storage systems.

Typically, the conventional SoC estimation approaches rely on a limited dataset collected under specific conditions, which might not accurately represent the full range of operating conditions for a battery [4]. Besides, they also often depend on physics-based models, which can be computationally intensive and sensitive to parameters variations such as hybrid Coulomb counting/impedance measurement approach [5]. To overcome these limitations and improve SoC estimation accuracy, machine learning techniques have emerged as promising alternatives. The rapid progress of contemporary machine learning techniques is driven by the continuous enhancement of computational capabilities and greater availability of datasets. Currently, machine learning algorithms have become deeply embedded in our daily life where they have emerged as the dominant choice for various tasks such as predictive analytics with particular attention to SoC estimation [6].

In reference to [7], a study conducted a comparison of 18 distinct

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

https://doi.org/10.1016/j.fraope.2023.100053
Received 28 August 2023; Received in revised form 15 October 2023; Accepted 14 November 2023
Available online 15 November 2023

machine learning techniques for estimating the SoC in batteries. In this study, machine learning techniques were categorized into three groups viz. linear models, ensemble models and other models. Within the linear model's category, it included Linear Regression, Bayesian Ridge, and a few others. Meanwhile, Bagging, XGBoost and several other methods were grouped under the ensemble methods category, while techniques like Support Vector Regression, Artificial Neural Network and a few others were grouped the under other models' category. The efficiency of these models was evaluated based on several criteria, such as training and prediction time comparisons, matching of SoC estimation curves, statistical evaluation, and performance indices. Among these findings, it becomes evident that the ensemble method outperforms the other groups, with Bagging and ExtraTree yielding superior results. An additional proposition found in [8] introduces the use of the Adaptive Neuro Fuzzy Inference System (ANFIS).

Meanwhile, in [9], a study proposed the use of Bidirectional Long Short Term Memory (BiLSTM)-Recurrent Neural Network (RNN) in conjunction with Parallel Artificial Neural Networks (PANNs) for estimating SoC in electric vehicle battery packs. The proposed model, implemented using a publicly available dataset, demonstrated superior performance compared to conventional RNNs, achieving a speedup of 1.5 −3 times. Ref. [10] presents a study that utilizes an LSTM-based model. In the proposed research, the LSTM model is enhanced with anti-noise adaptability to improve the accuracy of predicting remaining useful life (RUL). Another improved LSTM is introduced in [11]. Progressing further, a Deep Neural Networks (DNN) was introduced for the purpose of estimating SoC using data collected in a laboratory setting involving a Li-ion battery 18,650 [12]. When compared to a Gated Recurrent Unit Recurrent Neural Networks (GRU-RNN), the DNN network exhibited promising results. It yielded mean error values of less than 0.4 % and a maximum error value of under 2.5 % when applied to simulated data. These outcomes strongly indicate the accuracy of the DNN in the SoC estimation.

In recent study, Do et al. (2023) [13] introduced a hybrid model that merges the capabilities of the Extreme Learning Machine (ELM) model with Salp-Swarm Algorithm (SSA) [14] for the estimation of lithium-ion battery behavior. Given the ELM's sensitivity to network weights and hidden layer biases, the SSA is employed to search for the optimal values, while the ELM network is tailored to estimate the current SoC of the battery. The SSA's effectiveness is enhanced by incorporating a chaotic mapping technique during the initialization phase, and the Sine Cosine Algorithm (SCA) is integrated into the formulation of swarm positions. Through minimizing errors, the proposed model demonstrates its superiority when compared to other existing models, which includes Back Propagation Neural Network (BPNN), hybrid ELM with Particle Swarm Optimization (PSO) [15], among others. Similar works that demonstrate a hybrid machine learning with optimization algorithms for SoC estimation also can be seen in [16,17]. In addition, SoC prediction utilizing Kalman Filter has garnered significant attention in the research community, as elaborated upon in [18,19].

Among the reviewed existing works, Deep Learning (DL) has demonstrated remarkable capabilities in modeling complex relationships in data and solving challenging prediction tasks [20], including in battery SoC estimation [21]. DL models have predominantly found application in addressing issues related to classification, regression, and clustering tasks [22,23]. In contrast to conventional machine learning models such as Support Vector Machines (SVM) [24], Random Forest (RF), Logistic Regression (LR), among others, DL models depend on intricate neural networks known as deep neural network. These networks consist of multiple layers of neurons. DL has the ability to automatically extract representation from high dimensional data and acquire an understanding of complex nonlinear mapping between inputs and outputs [25]. In many instances, data driven approaches utilize the battery's measured current, voltage, and temperature as inputs, often employing a sliding window technique. Additionally, it is evident that DL can accommodate a noticeably larger sliding window length compared to traditional machine learning techniques.

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. To address this optimization challenge, this paper proposes a hybrid approach that integrates the Barnacles Mating Optimizer (BMO) [26] with DL (termed as BMO-DL) for SoC estimation in batteries. The BMO is a recent addition to the field of metaheuristic optimization. It draws inspiration from the mating behavior of barnacles in nature, aiming to replicate and adapt these principles to solve optimization problems effectively. The BMO has been proven its efficiency in solving complex and multi-dimensional optimization problems in different areas which includes in telecommunication networks [27], information security [28], power system [29,30], building system engineering [31], big-data [32], finance [33], electrical engineering [34,35], renewable energy [36,37], and many more. By hybridizing the BMO with DL, the strengths of both techniques can be maximized and achieve synergistic enhancement in addressing the issue of interest.

The contributions of this study are as follow:

i Automated hyper-parameter optimization of DL using BMO:

This study introduces a novel approach by using BMO to optimize the hyper-parameters of DL models. By automating the hyper-parameter tuning process, this contribution enhances the efficiency and effectiveness of DL for SoC estimation.

i Enhanced SoC estimation accuracy through hybrid BMO-DL

The hybrid BMO-DL presents a significant advantage in SoC estimation. By synergistically combining the strengths of both techniques, this contribution achieved a remarkably improvement in the SoC estimation.

The subsequent sections of this paper are organized as follows: In Section 2. A concise overview of barnacles in nature and the development of their mathematical model is presented, followed by a description on Deep Leaning 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 Section 5, and Section 6offers the concluding remarks.

2. Barnacles mating optimizer

2.1. Barnacles, in nature

Dating back to the Jurassic era, barnacles start their life cycle as swimmers and then adhere to aquatic surfaces, undergoing shell formation in adulthood. With a vast diversity of over 1400 species, barnacles are predominantly hermaphroditic, equipped with both male and female reproductive organism. Among them, acorn barnacles are particularly common. A striking trait of barnacles in their extended reproductive organ, setting a remarkable record in the animal kingdom relative to their body size [38]. Based on the uniqueness of the barnacles, the BMO was introduced which offers several advantages that make it a valuable choice for optimizing the weights and biases of Deep Learning.

i Exploration and Exploitation: BMO exhibits a balance between exploration and exploitation in the optimization process. This balance is crucial for deep learning, as it allows the algorithm to explore a wide range of potential solutions (exploration) while also fine-tuning promising solutions (exploitation). This property helps prevent getting stuck in local optima and can lead to finding better-performing neural network configurations.

ii Diversity in Search: The mating behavior modelled in BMO introduces diversity in the search process. In deep learning, having a diverse set of weight and bias configurations can lead to improved generalization and model robustness. BMO's ability to generate diverse candidate solutions aligns with the need for exploring a broad solution space in neural network optimization.

iii Global Optimization: BMO has demonstrated effectiveness in global optimization tasks. In Deep Learning, finding globally optimal weight and bias configurations is often crucial for achieving state-of-the-art performance.

2.2. Mathematical model of barnacles mating optimizer

The BMO consists of 3 phases, namely initialization, followed by selection, and finally, reproduction.

i Initialization

Initially, the number of possible solutions is initialized, which represents the barnacles. It can be expressed as follows:

$$X = \begin{bmatrix} x_1^1 & \cdots & x_1^N \\ \cdots & \cdots & \cdots \\ x_n^1 & \cdots & x_n^N \end{bmatrix} \begin{matrix} Barnacle/possible\ \ solution\ 1 \\ \cdots \\ Barnacle/possible\ \ solution\ n \end{matrix} \tag{1}$$

$$\overset{\rightarrow}{Problem\ \ dimension,N}$$

where;

N=number of control variables/optimization parameters

n= population size

The control variables outlined Eq. (1) are constraint with the upper and lower limits specific to the problem at hands, as defined below:

$$upperB = [upperB1, \ldots, upperBi] \tag{2}$$

$$lowerB = [lowerB1, \ldots, lowerBi] \tag{3}$$

where *upperB* and *lowerB* represent the upper and lower bounds of control variables, and i indicates the maximum number of control variables.

i Selection

The parameter pl governs the choice of pairing two barnacles for mating, inspired by the natural behavior of barnacles. This selection process operates under the following assumptions:

a The selection is random, but it is limited to the *pl* parameter.

b Each barnacle can contribute or receive sperm from other barnacles, with fertilization occurring between only one pair. Despite the possibility of multiple males fertilizing a female in nature [38], this model assumes a one-to-one fertilization.

c If the selection process picks the same barnacle, it implies self-mating (self-fertilization). However, self-mating is a rare occurrence among barnacles according to [39]. In these cases, the potential for self-mating is ignored, resulting in no new offspring generation.

d When the selection during a particular iteration exceeds the predetermined *pl* value, the process of sperm casting takes place, which is recognized as exploration.

Based on the information provided, exploitation corresponds to points (1) and (2).

To elaborate on the mating process, let's suppose that *pl* equals 7. As a result, in a given iteration, barnacle #1 has the opportunity to engage in mating only with barnacles ranging from barnacle #2 to barnacle #7. However, if barnacle #1 were to hypothetically choose barnacle #8, this would exceed the predetermined limit, leading to the absence of a mating event. In such cases, when this situation occurs, the process shifts into sperm casting, signifying an exploration phase. In this context, a straightforward selection procedure is employed, as follow:

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

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

where, *randperm* is short for "random permutation," is a function commonly used in optimization algorithms and other computational tasks. It generates a random permutation of a sequence of numbers, typically integers, from 1 to a specified maximum value. In BMO, this function is employed to shuffle the order in which elements are processed, introducing randomness and diversifying the exploration of potential solutions, thereby helping to avoid getting trapped in local optima and potentially discovering improved solutions. Details regarding the selection and mating process of the barnacles can be referred in [26].

i Reproduction

The subsequent formulas are employed to generate new offspring/ variables based on the barnacles' parents:

$$x_i^{N\_new} = px_{barnacle\_d}^N + qx_{barnacle\_m}^N \tag{6}$$

where p represents normally distributed pseudo-random numbers within the range of [0, 1], and q equals (1-p). The variables $x_{barnacle\_d}^N$ and $x_{barnacle\_m}^N$ correspond to the traits of the parent barnacles, Dad and Mum, respectively, as selected in Eqs. (4) and (5). The parameters p and q denote the proportions of characteristics inherited from *Dad* and *Mum* that contribute to the creation of new offspring. Thus, the offspring's traits are determined based on the probability of a random number falling between 0 and 1. For instance, if p is randomly generated as 0.6, it signifies that 60 % of the paternal (Dad) attributes are inherited, while 40 % of the maternal (Mum) attributes contribute to the formation of the new offspring.

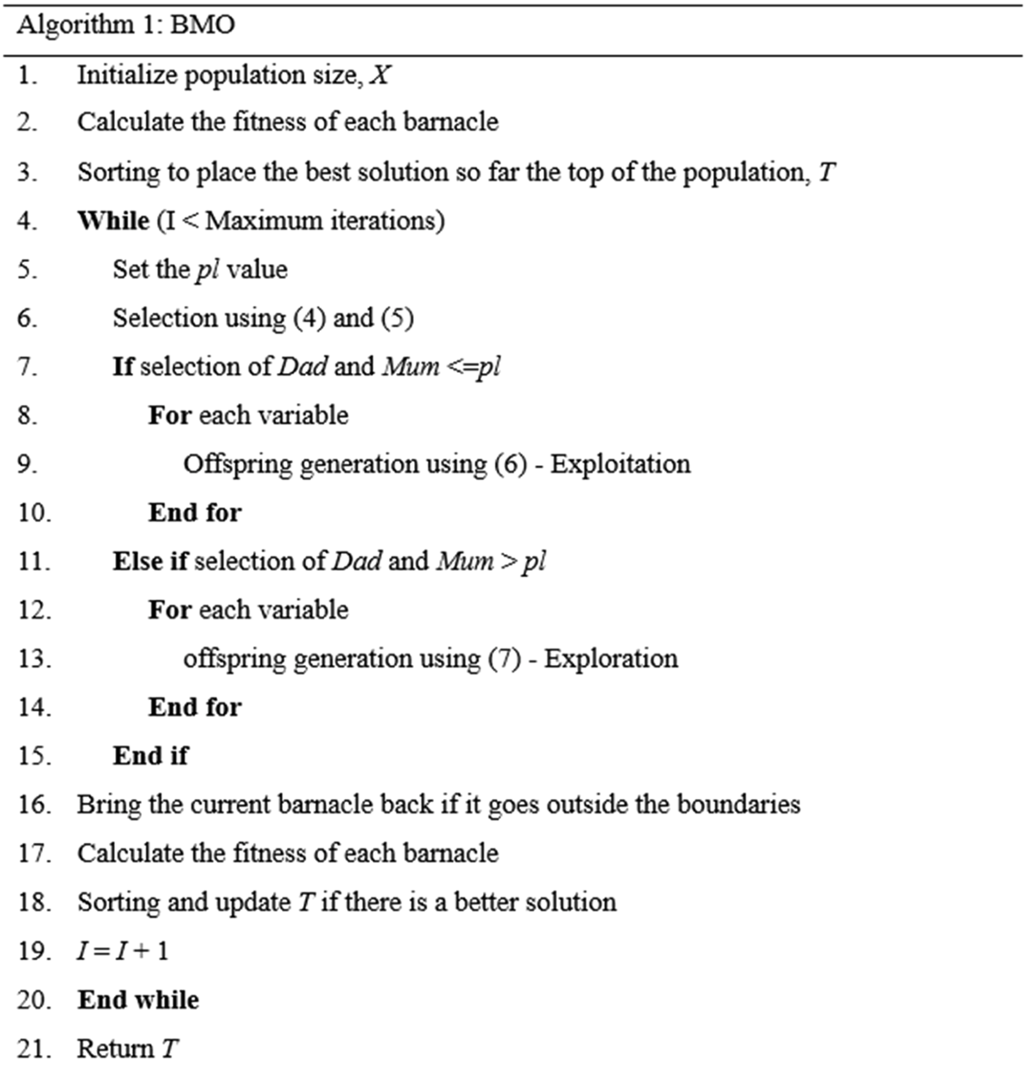
The parameter *pl* plays a crucial role in determining the equilibrium between the exploitation and exploration phases. Once the selection of barnacles for mating occurs within the *pl* range, it triggers the exploitation process (as indicated by Eq. (6)). On the other hand, if the selection falls beyond this range, the exploration proceed (Eq. (7)) is triggered, characterized as follow:

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

where *barnacle_d* and *barnacle_m* represent the potential parents for mating, and *n* denotes the population size. According to Eqs. (4) and (5), the selection process occurs randomly, aligning with the aforementioned assumption number 1. Here, *rand*() represents a random number within the range of [0,1]. The algorithm for BMO is depicted in Fig. 1.

3. Deep learning

For SoC estimation tasks, a DL is employed. A DL is feedforward, supervised learning network where in this study, it consists of an input layer, two hidden layers with 5 hidden neurons and an output layer. In input layer, the inputs are voltage (*V*), current (*I*) and conducted charge (*Q*) while the output is SoC in percentage. Due to the significant reliance of Deep Learning (DL) on the specific values of weights and biases, this research deviates from utilizing the Back Propagation (BP) algorithm for network training. Instead, the study adopts the BMO approach (refer to Sections 3 and 4.2). The architecture of DL for State of Charge (SoC) estimation is illustrated in Fig. 2.



```
Algorithm 1: BMO
Initialize population size, $X$
Calculate the fitness of each barnacle
Sorting to place the best solution so far the top of the population, $T$
While (I < Maximum iterations)
    Set the $pl$ value
    Selection using (4) and (5)
    If selection of $Dad$ and $Mum$ <=$pl$
        For each variable
            Offspring generation using (6) - Exploitation
        End for
    Else if selection of $Dad$ and $Mum$ > $pl$
        For each variable
            offspring generation using (7) - Exploration
        End for
    End if
Bring the current barnacle back if it goes outside the 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. 1. BMO Algorithm.

4. Methodology

This section is dedicated to elaborating on the methodology utilized in this study. It encompasses a comprehensive description of the dataset, the procedures for training and testing, the hybrid BMO-DL model, and finally, the criteria employed for evaluation.

4.1. Dataset description

In this study, the data was gathered through simulations involving an electric car resembling the Nissan Leaf, utilizing a lithium polymer cell model known as ePLB C020 [40]. The dataset incorporated information about the battery, including V, I and Q, which are fed to the estimation model as inputs. The datasets consist of 68,741 instances which are later divided into training and testing set. Sample of dataset are as tabulated in Table 1 [40]:

Meanwhile, Fig. 3 illustrates the holistic input-output dataset employed for the training process. This dataset comprises a substantial collection of over 40 thousand instances, which translates to more than 60 % of the complete dataset. Notably, this dataset showcases the raw data in its original form before undergoing the normalization procedure. The utilization of raw data is crucial, as it provides insights into the inherent characteristics and distribution of the information. Subsequently, the normalization process transforms this raw data into a standardized format, which is essential for enhancing the model's learning process and promoting convergence during training. This practice ensures that the neural network effectively learns from a consistent and comparable range of values, facilitating improved performance and accurate predictions.

4.1.1. Training and testing

To facilitate both training and testing, a duration of 12 h was allocated for training, encompassing a trip of 277.64 km. Meanwhile, for testing, a separate 7-hour dataset was employed, corresponding to a trip of 163.24 km. The distribution for training and testing was divided in the ratio of 0.6 and 0.4, respectively.

4.2. Barnacles mating optimizer-deep learning

In this research, the BMO algorithm was applied to enhance the performance of the DL by fine-tuning its weights and biases. This optimization process seamlessly integrated the BMO algorithm into the DL framework, functioning until the predefined maximum iteration limit is attained. For this research, the maximum iteration is established at 1000. The optimization process centered on key parameters within the neural network, specifically focusing on the weights denoted as w_{ji} and w_{kj}, as well as the biases situated in both the hidden and output layers, which bounded between -1 to 1. The setup comprises a total of 5 hidden neurons, which have been determined through iterative experimentation. The total variables to be optimized (weights and biases, Fig. 2 as

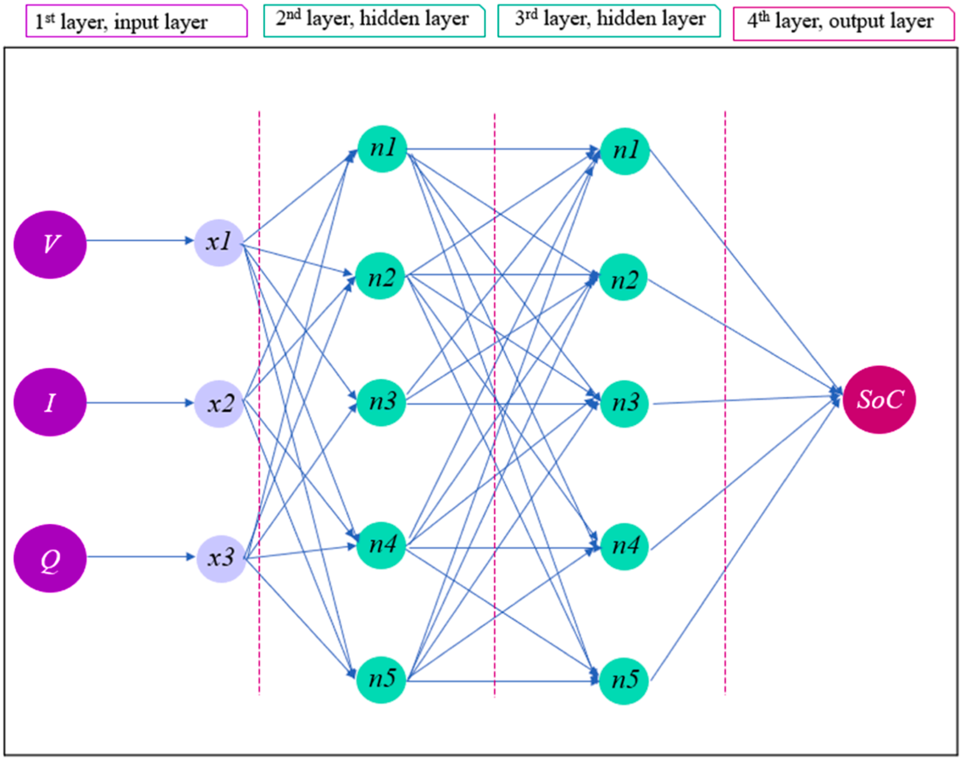


Fig. 2. Deep Learning Architecture for SoC estimation.

Table 1
Sample of Dataset.

Sampling Time (s)	V	C	Q
1	4.1749	0.0090	0
2	4.1749	0.0090	0
3	4.1748	−0.0090	−2.6455
4	4.1703	−0.3797	−1.0744
5	4.1697	−0.3706	−2.1136
6	4.1695	−0.3526	−3.0839
7	4.1694	−0.3390	−4.0317
8	4.1694	−0.3028	−4.8701
9	4.1692	−4.2217	−5.8377
10	4.1104	−7.9959	−0.0017

reference) are 3 (inputs neuron) x 5 (hidden neuron layer 1) + 5 biases (hidden neuron layer 1) + 5 (hidden neuron layer 1) x 5 (hidden neuron layer 2) + 5 biases (hidden neuron layer 2) + 5 (hidden neuron layer 2) x 1 (output neuron) + 1 bias (output neuron) = 56 variables. The primary objective function utilized is the MAE. Significantly, the goal was to minimize the MAE as it corresponds to superior model performance and predictive accuracy. To provide readers with a visual representation of the BMO-DL optimization process, a flowchart of BMO-DL has been included in Fig. 4. These comprehensive adjustments ensure that the DL model attains an optimal configuration of weights and biases within the specified iteration limit, ultimately contributing to an enhanced model performance and improved predictive accuracy.

4.3. Performance evaluation metrics

In this study, the evaluation of the SOC estimation is served by two criteria, namely Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). AME provides the average of the absolute differences between the estimated values and the target values. It measures the average magnitude of errors without considering their direction, making it less sensitive to outliers. Meanwhile, RMSE is the square root of the average of the squared differences between the estimated values and the target values. It places more emphasis on larger errors due to the squaring of errors, making it sensitive to outliers. Both metrics are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \| y(i) - \widehat{y}(i) \|^2}{N}} \tag{8}$$

$$MAE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} |y(i) - \widehat{y}(i)|} \tag{9}$$

where N represents the data length of the battery estimated to be evaluated, y(i) and $\hat{y}$(i) are the target and estimated battery SOC, respectively.

4.4. Benchmarking technique

This study undertakes a comparison of the results attained through the BMO-DL approach with those achieved by optimizing DL using the Particle Swarm Optimization (PSO) method and the Harmony Search Algorithm (HSA). A brief of PSO and HSA are as follow:

4.4.1. PSO

Introduced by Kennedy and Eberhart [13], PSO was developed based on the inspiration of birds' movements or fish schooling. Classified under Swarm Intelligence (SI) algorithm, PSO is based on the principle that the members of the swarm are cooperative with each other. In PSO, each particle possesses its individual best solution (Pb), as well as global best solution (Gb). The Gb represents the most optimal solution among all the Pb achieved so far. Each particle's behavior is influenced by exchanging information with other particles within the swarm. The continuous evolution of each particle's trajectory is captured by adjusting its velocity and position using the defined equations, respectively. Since its introduction, it has been widely used to solve various problems in different fields [41–43].

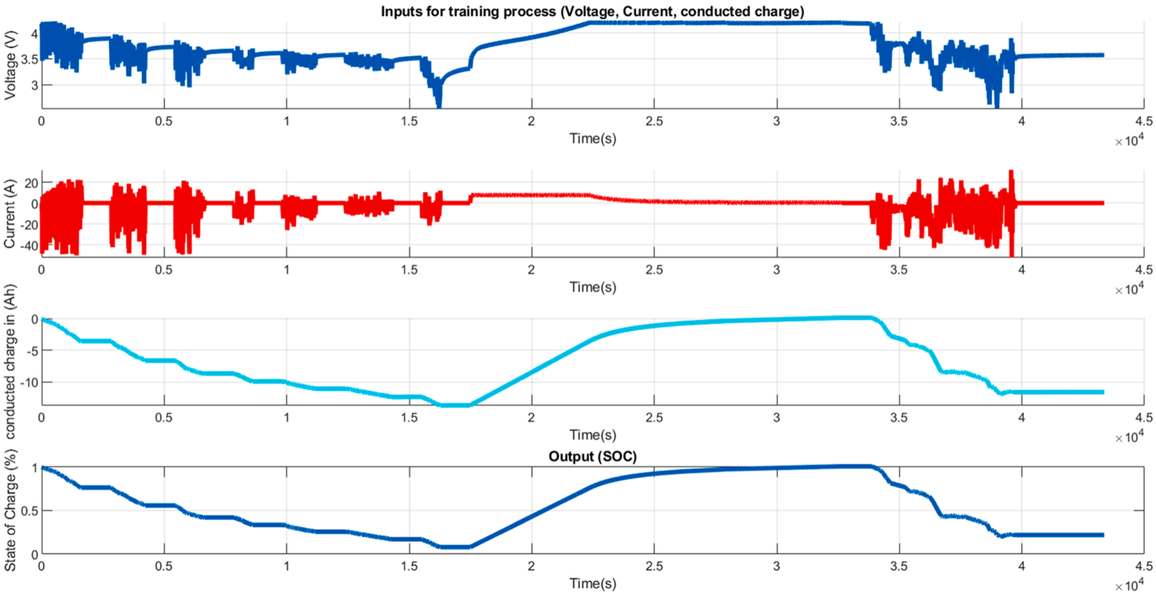


Fig. 3. Input-Output data for training process.

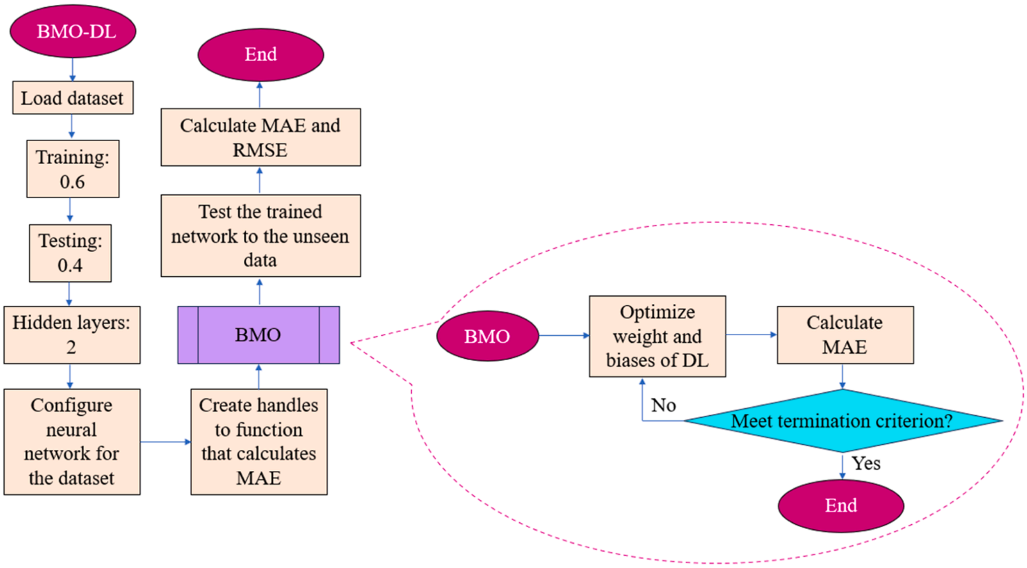

BMO-DL
Load dataset
Training: 0.6
Testing: 0.4
Hidden layers: 2
Configure neural network for the dataset
Create handles to function that calculates MAE
BMO
Test the trained network to the unseen data
Calculate MAE and RMSE
End
BMO
Optimize weight and biases of DL
Calculate MAE
Meet termination criterion?
No
Yes
End

Fig. 4. Flowchart of BMO-DL.

4.4.2. HSA

Unlike PSO, HSA [44] is categorized as human-inspired algorithm, which draws inspiration from the musical process of finding harmonious arrangements of musical notes and applies that concept to optimization problems. In HAS, potential solutions are represented as harmonies within a harmony memory (HM). These harmonies undergo a transformation process that mirrors the improvisational creation of new harmonies by musicians. This process involves generating a novel harmony by following three pitch production rules: selecting a pitch at random from HM, adjusting a randomly chosen pitch from HM, and constructing a pitch through random selection. If the newly generated harmony surpasses the quality of the weakest harmony within HM, it replaces the latter. This iterative process persists until a specified termination condition is met. The application of HSA can be seen in various fields such as scheduling [45,46], education [47] and many more.

5. Results and discussion

In order to identify the optimal configurations of hidden neurons for BMO-DL, the proposed model underwent experimentation using three distinct settings for hidden neurons, as outlined in Table 2. Besides MAE and RMSE, the evaluation also takes into account the maximum error which represents the largest difference between predicted and actual values. A lower maximum error indicates that the method produces fewer extreme outliers.

Based on Table 2, the collected results show a performance

Table 2
Comparison of Different Hidden Neurons for BMO-DL.

Number of Hidden Neuron	MAE	RMSE	Maximum Error
3	8.7293	10.4943	17.4227
5	5.6610	6.7888	12.3972
7	8.3934	9.6334	14.2611

comparison among various configurations of hidden neurons. Three key metrics were employed namely MAE, RMSE and maximum error. For the case of hidden neurons=3, the recorded MAE, RMSE and maximum error are 8.7293, 10.4943 and 17.4227, respectively. Moving to the configuration with five hidden neurons, the metrics exhibit a much smaller MAE, which is 5.6610 and 6.7888 of RMSE while the maximum error was recorded at 12.3972. Finally, for the setup involving seven hidden neurons, the values showcase a MAE of 8.3934, RMSE of 9.6334, and a maximum error of 14.2611.

The tabulated findings provide insights into the impact of varying hidden neuron counts on the performance of BMO-DL estimation model. As evident from the above table, the configuration employing five hidden neurons seems to produce the smallest MAE and RMSE values. However, it is noting that this configuration also demonstrates a relatively higher maximum error. Conversely, the three-hidden-neuron configuration and the seven-hidden-neuron configuration have their respective strengths and weaknesses in terms of estimation error. Fig. 5 illustrates the performance comparison of different hidden neurons for BMO-DL. This comparison is based on three key metrics: MAE, RMSE, and Maximum Error. To ensure the robustness and reliability of the results, simulations were conducted for each hidden neuron configuration (3, 5, and 7) independently and repeatedly. This approach was adopted to account for potential variability in the outcomes and to assess the consistency of the model's performance. Upon analyzing the results of these simulations, a consistent pattern emerged. The configuration with 5 hidden neurons consistently outperformed the others across all five runs. Thus, the configuration with 5 neurons at each hidden layers is selected for the BMO-DL for the SoC estimation problem.

Table 3 shows the result of the performance comparison among BMO-DL, PSO-DL and HAS-DL. Considering the data provided in the table above, it becomes clear that BMO-DL exhibits the smallest MAE value, indicating its superior performance in minimizing the average error. This trend is consistent in the RMSE results as well, where BMO-DL outperforms the other methods by having the lowest RMSE. This signifies that BMO-DL offers more accurate predictions across the board. Furthermore, when considering the metric of maximum error, the results obtained by BMO-DL highlight its greater consistency in avoiding extremely inaccurate predictions. The HSA-DL secured the second position, exhibiting higher MAE and RMSE values than those achieved by BMO-DL, specifically 10.0983 and 10.6445, respectively. This places PSO-DL in the third position. The maximum errors produced by all identified algorithms seem to align with the recorded errors.

Figs. 6–8 depict the estimated values generated by BMO, PSO and HSA, respectively, in comparison with the actual SoC values. The lower portion of each figure visualizes the errors associated with the estimations produced by their respective method.

To enhance the clarity of the comparison, we have aggregated the results from Figs. 6 to 8, as illustrated in Fig. 9. As seen in the figure, the values generated by BMO-DL closely align with the target values, especially at the onset of the testing phase. In contrast, the prediction values generated by PSO-DL appear to be scattered throughout most of the testing phase.

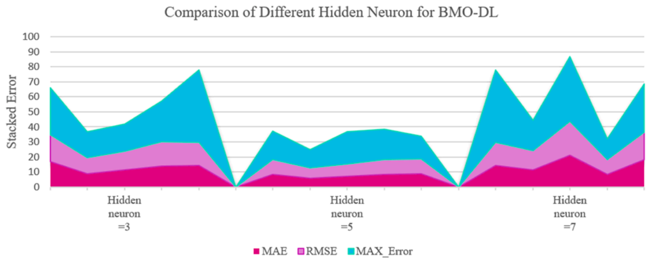


Fig. 5. Performance comparison of different hidden neurons for BMO-DL.

Table 3
Comparison between BMO-DL vs. PSO-DL vs. HAS-DL for SoC Estimation.

Hybrid Methods/Metrics	MAE	RMSE	Maximum Error
BMO-DL	5.6610	6.7888	12.3972
PSO-DL	13.3401	14.1219	22.7693
HSA-DL	10.0983	10.6445	16.7805

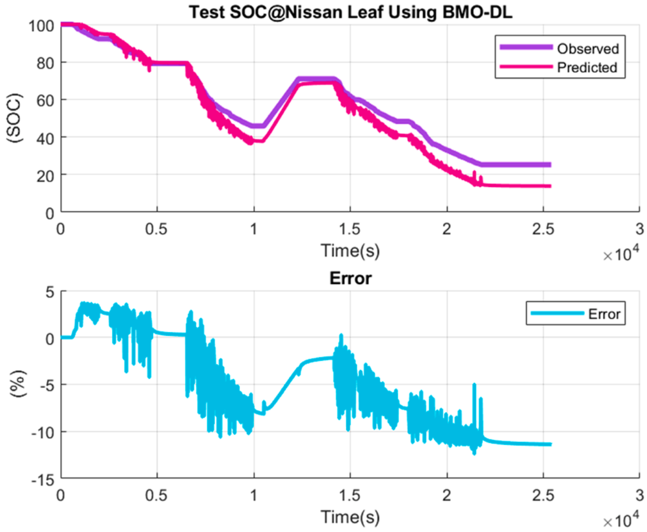


Fig. 6. SoC Estimation based on BMO-DL.

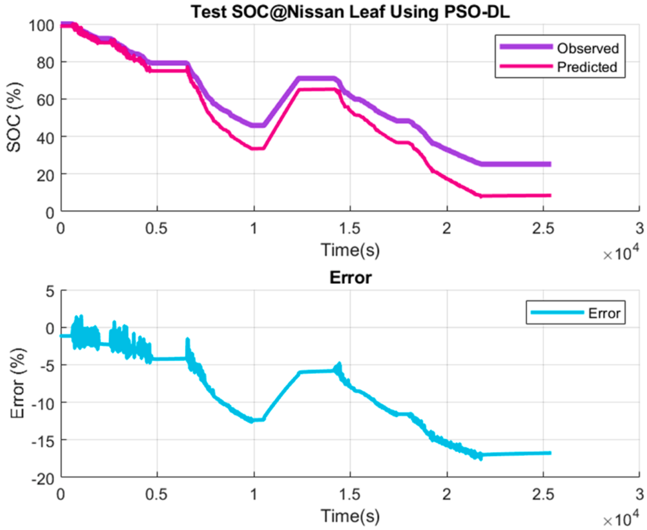


Fig. 7. SoC Estimation based on PSO-DL.

Fig 10. illustrates the comparison of convergence curve recorded by BMO-DL, PSO-DL and HSA-DL. The convergence performance was evaluated over a total of 1000 iterations. Among these methods, BMO-DL exhibited a notably superior convergence rate, achieving a MAE value of 0.0079. In comparison, the PSO-DL method displayed slightly competitive convergence capabilities, yielding an MAE value of 0.0085. In contrast, the HSA-DL demonstrated slightly higher convergence values, recording an MAE of 0.0241. These results highlight the robust convergence performance of BMO-DL which converges to solutions with remarkable precision over the course of 1000 iterations.

A significant test for each difference with the identified methods is provided in Table 4. From the table, it is evident that the differences in performance among these methods are statistically significant. These results suggest that BMO-DL, in both comparisons, demonstrates a clear

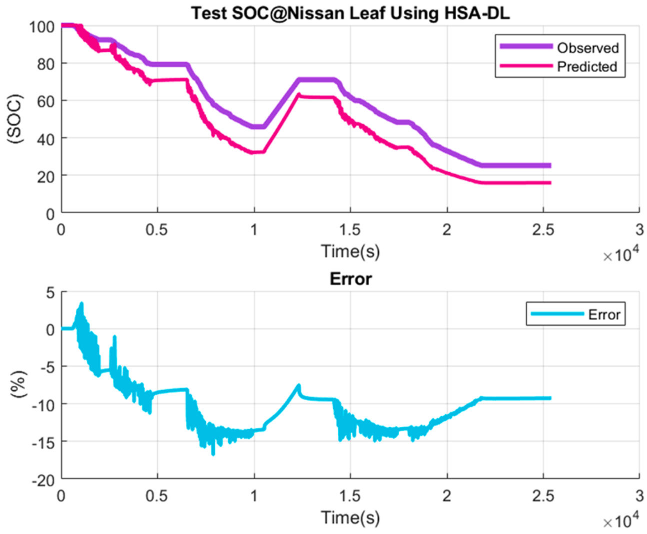


Fig. 8. SoC Estimation based on HSA-DL.

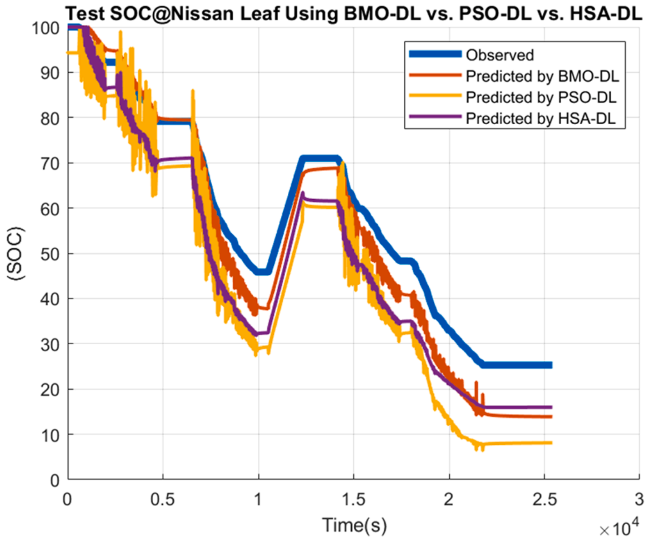


Fig. 9. Comparison of BMO-DL vs. PSO-DL vs. HSA-DL for SoC Estimation.

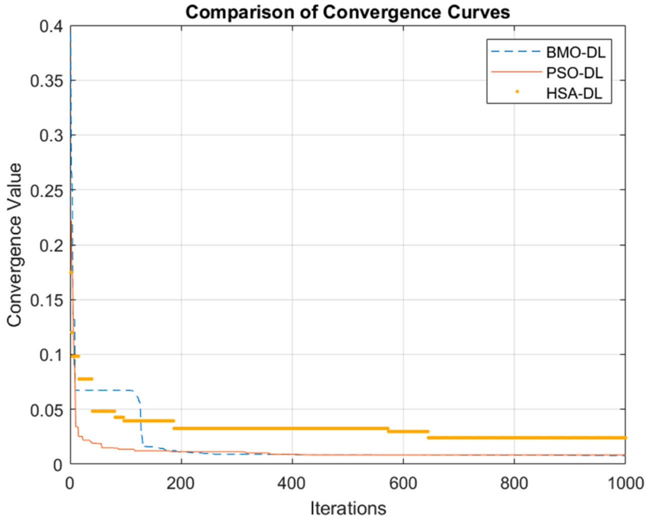


Fig. 10. Comparison of Convergence Values.

Table 4
Significant Test for SoC Estimation: BMO-DL vs. PSO-DL and HSA-DL.

Methods	p(*T*<=*t*) two tail
BMO-DL – PSO-DL	0.000
BMO-DL – HAS-DL	0.000

advantage over the other methods in terms of accuracy, as indicated by the highly significant *p*-values.

6. Conclusion

In conclusion, this study introduced the BMO-DL hybrid approach for precise SoC estimation in battery systems. The methodology combined BMO with DL and underwent rigorous evaluation on Nissan Leaf batteries. Using a comprehensive dataset, the performance of the BMO-DL model was assessed with two key metrics: MAE and RMSE. Additionally, a comparative analysis was conducted to establish the statistical significance of BMO-DL's outcomes in relation to two analogous hybrid techniques, namely hybrid PSO-DL and HSA-DL.

The study's findings consistently demonstrated that BMO-DL outperformed the identified hybrid algorithms, yielding significantly lower MAE and RMSE values. This highlights the potential of BMO-DL as an effective approach for accurate SoC estimation in battery systems. In summary, the BMO-DL hybrid approach represents a notable advancement in battery SoC estimation. Its superior performance compared to existing hybrid algorithms suggests practical applicability in electric vehicles and renewable energy systems. As the research landscape continues to evolve, this study provides a foundation for future endeavors, driving further innovations and advancements in battery management and estimation techniques.

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.

Acknowledgements

This work is supported by Universiti Malaysia Pahang Al-Sultan Abdullah research grant (RDU220379).

References

[1] H. Lu, X. Chen, C. Fang, H. Yang, Data analytics for optimizing extreme fast charging: a survey, Data Sci. Manage. 1 (1) (2021) 23–31, https://doi.org/10.1016/j.dsm.2021.02.001.

[2] G. Tian, et al., Recycling of spent Lithium-ion Batteries: a comprehensive review for identification of main challenges and future research trends, Sustain. Energy Technol. Assess. 53 (2022), 102447, https://doi.org/10.1016/j.seta.2022.102447.

[3] E. Chemali, P.J. Kollmeyer, M. Preindl, A. Emadi, State-of-charge estimation of Li-ion batteries using deep neural networks: a machine learning approach, J. Power Sour. 400 (2018) 242–255, https://doi.org/10.1016/j.jpowsour.2018.06.104.

[4] S. D.V.S.R, C. Badachi, R.C. Green Ii, A review on data-driven SOC estimation with Li-Ion batteries: implementation methods & future aspirations, J. Energy Storage 72 (2023), 108420, https://doi.org/10.1016/j.est.2023.108420.

[5] M. Becherif, H.S. Ramadan, A. Benmouna, S. Jemei, Initial state of charge estimation of battery using impedance measurement for electrical vehicle applications, Sustain. Energy Technol. Assess. 53 (2022), 102727, https://doi.org/10.1016/j.seta.2022.102727.

[6] P. Vasanthkumar, A.R. Revathi, G. Ramya Devi, R.J. Kavitha, A. Muniappan, C. Karthikeyan, Improved wild horse optimizer with deep learning enabled battery management system for internet of things based hybrid electric vehicles, Sustain. Energy Technol. Assess. 52 (2022), 102281, https://doi.org/10.1016/j.seta.2022.102281.

[7] M. Korkmaz, SoC estimation of lithium-ion batteries based on machine learning techniques: a filtered approach, J. Energy Storage 72 (2023), 108268, https://doi.org/10.1016/j.est.2023.108268.

[8] R. Vignesh, B. Ashok, M. Senthil Kumar, D. Szpica, A. Harikrishnan, H. Josh, Adaptive neuro fuzzy inference system-based energy management controller for

optimal battery charge sustaining in biofuel powered non-plugin hybrid electric vehicle, Sustain. Energy Technol. Assess. 59 (2023), 103379, https://doi.org/10.1016/j.seta.2023.103379.
[9] A. Manoharan, D. Sooriamoorthy, K.M. Begam, V.R. Aparow, Electric vehicle battery pack state of charge estimation using parallel artificial neural networks, J. Energy Storage 72 (2023), 108333, https://doi.org/10.1016/j.est.2023.108333.
[10] S. Wang, Y. Fan, S. Jin, P. Takyi-Aninakwa, C. Fernandez, Improved anti-noise adaptive long short-term memory neural network modeling for the robust remaining useful life prediction of lithium-ion batteries, Reliab. Eng. Sys. Saf. 230 (2023), 108920, https://doi.org/10.1016/j.ress.2022.108920.
[11] S. Wang, F. Wu, P. Takyi-Aninakwa, C. Fernandez, D.-I. Stroe, Q. Huang, Improved singular filtering-Gaussian process regression-long short-term memory model for whole-life-cycle remaining capacity estimation of lithium-ion batteries adaptive to fast aging and multi-current variations, Energy 284 (2023), 128677, https://doi.org/10.1016/j.energy.2023.128677.
[12] S. El Fallah, J. Kharbach, Z. Hammouch, A. Rezzouk, M. Ouazzani Jamil, State of charge estimation of an electric vehicle's battery using deep neural networks: simulation and experimental results, J. Energy Storage 62 (2023), 106904, https://doi.org/10.1016/j.est.2023.106904.
[13] J. Dou, et al., Extreme learning machine model for state-of-charge estimation of lithium-ion battery using salp swarm algorithm, J. Energy Storage 52 (2022), 104996, https://doi.org/10.1016/j.est.2022.104996.
[14] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Soft. 114 (2017) 163–191, https://doi.org/10.1016/j.advengsoft.2017.07.002.
[15] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, 27 Nov.-1 Dec. 1995 4, 1995, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968.
[16] M.H. Zafar, et al., Hybrid deep learning model for efficient state of charge estimation of Li-ion batteries in electric vehicles, Energy 282 (2023), 128317, https://doi.org/10.1016/j.energy.2023.128317.
[17] M. Li, C. Li, Q. Zhang, W. Liao, Z. Rao, State of charge estimation of Li-ion batteries based on deep learning methods and particle-swarm-optimized Kalman filter, J. Energy Storage 64 (2023), 107191, https://doi.org/10.1016/j.est.2023.107191.
[18] P. Reshma, V.J. Manohar, Collaborative evaluation of SoC, SoP and SoH of lithium-ion battery in an electric bus through improved remora optimization algorithm and dual adaptive Kalman filtering algorithm, J. Energy Storage 68 (2023), 107573, https://doi.org/10.1016/j.est.2023.107573.
[19] Rimsha, et al., State of charge estimation and error analysis of lithium-ion batteries for electric vehicles using Kalman filter and deep neural network, J. Energy Storage 72 (2023), 108039, https://doi.org/10.1016/j.est.2023.108039.
[20] R. Pugliese, S. Regondi, R. Marini, Machine learning-based approach: global trends, research directions, and regulatory standpoints, Data Sci. Manage. 4 (2021) 19–29, https://doi.org/10.1016/j.dsm.2021.12.002.
[21] B. Jiang, S. Tao, X. Wang, J. Zhu, X. Wei, H. Dai, Mechanics-based state of charge estimation for lithium-ion pouch battery using deep learning technique, Energy 278 (2023), 127890, https://doi.org/10.1016/j.energy.2023.127890.
[22] J. Liang, G. Jia, China futures price forecasting based on online search and information transfer, Data Sci. Manage. 5 (4) (2022) 187–198, https://doi.org/10.1016/j.dsm.2022.09.002.
[23] D. Lasantha, S. Vidanagamachchi, S. Nallaperuma, Deep learning and ensemble deep learning for circRNA-RBP interaction prediction in the last decade: a review, Eng Appl. Artif. Intell. 123 (2023), 106352, https://doi.org/10.1016/j.engappai.2023.106352.
[24] V.N. Vapnik, The Nature of Statistical Learning Theory, 2nd ed., Springer-Verlag, New York, 1995.
[25] H. Liu, Z. Fan, J. Lin, Y. Yang, T. Ran, H. Chen, The recent progress of deep-learning-based in silico prediction of drug combination, Drug Discov. Today 28 (7) (2023), 103625, https://doi.org/10.1016/j.drudis.2023.103625.
[26] 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.
[27] 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.
[28] 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.
[29] 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.
[30] M. Rawa, Y. Al-Turki, K. Sedraoui, S. Dadfar, M. Khaki, Optimal operation and stochastic scheduling of renewable energy of a microgrid with optimal sizing of battery energy storage considering cost reduction, J. Energy Storage 59 (2023), 106475, https://doi.org/10.1016/j.est.2022.106475.
[31] M.H. Sulaiman, Z. Mustaffa, Optimal chiller loading solution for energy conservation using barnacles mating optimizer algorithm, Results Control Optim. 7 (2022), 100109, https://doi.org/10.1016/j.rico.2022.100109.
[32] 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.
[33] 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/.
[34] B. Liu, H. Wang, M.-L. Tseng, Z. Li, State of charge estimation for lithium-ion batteries based on improved barnacle mating optimizer and support vector machine, J. Energy Storage 55 (2022), 105830, https://doi.org/10.1016/j.est.2022.105830, 2022/11/30/.
[35] H. Li, H. Guo, N. Yousefi, A hybrid fuel cell/battery vehicle by considering economy considerations optimized by Converged Barnacles Mating Optimizer (CBMO) algorithm, Energy Rep. 6 (2020) 2441–2449, https://doi.org/10.1016/j.egyr.2020.09.005, 2020/11/01/.
[36] R.M. Rizk-Allah, A.A. El-Fergany, Conscious neighborhood scheme-based Laplacian barnacles mating algorithm for parameters optimization of photovoltaic single- and double-diode models, Energy Convers. Manage. 226 (2020), 113522, https://doi.org/10.1016/j.enconman.2020.113522.
[37] G. Fan, M. Li, X. Chen, X. Dong, K. Jermsittiparsert, Analysis of a multi-objective hybrid system to generate power in different environmental conditions based on improved the barnacles mating optimizer algorithm, Energy Rep. 7 (2021) 2950–2961, https://doi.org/10.1016/j.egyr.2021.05.023.
[38] Marjan Baranzandeh, et al., Something Darwin didn't know about barnacles: spermcast mating in a common stalked species, Proc. R. Soc. B 280 (1754) (2013).
[39] Y. Yusa, et al., Adaptive evolution of sexual systems in pedunculate barnacles, Proc. R. Soc. B (2012) 959–966.
[40] M. Luzi. "Automotive Li-ion cell usage data set." IEEEDataPort. https://ieee-dataport.org/documents/automotive-li-ion-cell-usage-data-set (Accessed 1, August, 2023).
[41] S. Huan, A novel interval decomposition correlation particle swarm optimization-extreme learning machine model for short-term and long-term water quality prediction, J. Hydrol. (Amst) 625 (2023), 130034, https://doi.org/10.1016/j.jhydrol.2023.130034.
[42] M.P. Behera, A. Sarangi, D. Mishra, S.K. Sarangi, A hybrid machine learning algorithm for heart and liver disease prediction using modified particle swarm optimization with support vector machine, Procedia Comput. Sci. 218 (2023) 818–827, https://doi.org/10.1016/j.procs.2023.01.062.
[43] X. Zhao, et al., Exploring interpretable and non-interpretable machine learning models for estimating winter wheat evapotranspiration using particle swarm optimization with limited climatic data, Comput. Electron. Agricul. 212 (2023), 108140, https://doi.org/10.1016/j.compag.2023.108140.
[44] Zong Woo Geem, Joong Hoon Kim, G.V. Loganathan, A New Heuristic Optimization Algorithm: harmony Search, Simulation 76 (2) (2001) 60–68.
[45] E.A. Khan, M.K. Shambour, An optimized solution for the transportation scheduling of pilgrims in Hajj using harmony search algorithm, J. Eng. Res. 11 (2) (2023), 100038, https://doi.org/10.1016/j.jer.2023.100038.
[46] I. Abu Doush, M.A. Al-Betar, M.A. Awadallah, Z.A.A. Alyasseri, S.N. Makhadmeh, M. El-Abd, Island neighboring heuristics harmony search algorithm for flow shop scheduling with blocking, Swarm Evol. Comput. 74 (2022), 101127, https://doi.org/10.1016/j.swevo.2022.101127.
[47] R.H. Chae, A.C. Regan, An analysis of harmony search for solving Sudoku puzzles, Soft Comput. Lett. 3 (2021), 100017, https://doi.org/10.1016/j.socl.2021.100017.