Energy Storage and Saving 4 (2025) 111–122

Contents lists available at ScienceDirect

Energy Storage and Saving

journal homepage: www.keaipublishing.com/en/journals/energy-storage-and-saving/

Research article

State of charge estimation of lithium-ion batteries in an electric vehicle using hybrid metaheuristic - deep neural networks models

Zuriani Mustaffa [a,*], Mohd Herwan Sulaiman [b], Jeremiah Isuwa [c]

[a] Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), 26600, Pekan Pahang, Malaysia
[b] Faculty of Electrical & Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), 26600, Pekan Pahang, Malaysia
[c] Department of Computer Science, Federal University of Kashere, Kashere, 771103, Gombe, Nigeria

ARTICLE INFO

Keywords:
Deep learning
Deep neural networks
Electric vehicle
Machine learning
Optimization
State of charge estimation
Teaching-learning based optimization

ABSTRACT

Accurate estimation of the state of charge (SoC) of lithium-ion batteries (LIBs) in electric vehicles (EVs) is crucial for optimizing performance, ensuring safety, and extending battery life. However, traditional estimation methods often struggle with the nonlinear and dynamic behavior of battery systems, leading to inaccuracies that compromise the efficiency and reliability of electric vehicles. This study proposes a novel approach for SoC estimation in BMW EVs by integrating a metaheuristic algorithm with deep neural networks. Specifically, teaching-learning based optimization (TLBO) is employed to optimize the weights and biases of the deep neural networks model, enhancing estimation accuracy. The proposed TLBO-deep neural networks (TLBO-DNNs) method was evaluated on a dataset of 1,064,000 samples, with performance assessed using mean absolute error (MAE), root mean square error (RMSE), and convergence value. The TLBO-DNNs model achieved an MAE of 3.4480, an RMSE of 4.6487, and a convergence value of 0.0328, outperforming other hybrid approaches. These include the barnacle mating optimizer-deep neural networks (BMO-DNNs) with an MAE of 5.3848, an RMSE of 7.0395, and a convergence value of 0.0492; the evolutionary mating algorithm-deep neural networks (EMA-DNNs) with an MAE of 7.6127, an RMSE of 11.2287, and a convergence value of 0.0536; and the particle swarm optimization-deep neural networks (PSO-DNNs) with an MAE of 4.3089, an RMSE of 5.9672, and a convergence value of 0.0345. Additionally, the TLBO-DNNs approach outperformed standalone models, including the autoregressive integrated moving average (ARIMA) model (MAE: 14.3301, RMSE: 7.0697) and support vector machines (SVMs) (MAE: 6.0065, RMSE: 8.0360). This hybrid TLBO-DNNs technique demonstrates significant potential for enhancing battery management systems (BMS) in electric vehicles, contributing to improved efficiency and reliability in electric vehicle operations.

1. Introduction

Electric vehicles (EVs) have gained considerable attention in recent years as a means to reduce carbon emissions from road transportation, which accounted for 23% of global energy-related CO_2 emissions across economic sectors in 2019 [1]. Growing environmental concerns associated with internal combustion engine (ICE) vehicles have accelerated the transition to EVs as a viable strategy for reducing greenhouse gas emissions in the transportation sector. Data from the United States Department of Energy (DOE) highlight this shift, showing that conventional ICE vehicles produced approximately 4.5 times more annual CO_2 emissions (12,594 pounds in 2021) compared to all-electric vehicles (2,817 pounds in 2021). Notably, EVs running exclusively on electricity generate zero greenhouse gas emissions, including CO_2, methane (CH_4), and nitrous oxide (N_2O), during operation [2]. This trend reflects global efforts to mitigate climate change, driven by increasing concerns over the environmental impact of traditional combustion engines [3]. The transition from fossil fuel-powered vehicles to electric alternatives represents a critical step in reducing greenhouse gas emissions, which are a primary driver of global warming. While electricity generation has historically relied on centralized power plants, concerns over fossil fuel depletion and environmental sustainability have spurred the search for alternative, highly efficient energy production methods [4]. Governments worldwide are implementing policies and incentives to encourage

Peer review under responsibility of Xi'an Jiaotong University.

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

https://doi.org/10.1016/j.enss.2025.01.002
Received 21 October 2024; Received in revised form 30 November 2024; Accepted 15 January 2025
Available online 16 February 2025

EV adoption, such as tax credits, rebates, and investments in charging infrastructure. Furthermore, advancements in battery technology and increased production efficiency are making EVs more affordable and accessible to a broader range of consumers. This transition in the transportation sector is part of a broader strategy to electrify various systems traditionally reliant on fossil fuels, contributing to overall carbon reduction goals. As more renewable energy sources are integrated into the power grid, the environmental benefits of EVs are expected to increase, further supporting the transition to a more sustainable and resilient energy future.

The widespread adoption of EVs has led to a significant reliance on lithium-ion batteries (LIBs) [5], highlighting the importance of accurately estimating the state of charge (SoC). Proper SoC estimation is essential for ensuring battery reliability, safety, and longevity, as it helps manage energy usage and predicts the remaining driving range. Advances in battery management systems (BMS) and SoC estimation techniques are crucial for optimizing LIBs performance, enhancing driver convenience, and supporting the broader adoption of EVs. LIBs, known for their low self-discharge rate, high power density, and long cycle life, remain the leading choice for energy storage, further facilitating the shift toward sustainable transportation solutions [6]. These qualities have paved the way for advancements in the EVs market and driven the widespread adoption of portable electronic devices. Various SoC estimation methods, including Coulomb counting, model-based approaches, and data-driven techniques, have been developed to improve estimation accuracy. Coulomb counting estimates SoC by calculating the ratio of remaining charge to the total battery capacity [7]. While often combined with open-circuit voltage curves to mitigate self-discharge errors, this method struggles with error accumulation due to sensor inaccuracies. Model-based approaches, including empirical, electrical equivalent circuit, electrochemical, and electrochemical impedance models, as well as Kalman filter algorithms [8], help address current sensor measurement issues. However, their accuracy depends heavily on the precision of the battery model [9].

In recent decades, the implementation of machine learning models for SoC estimation has demonstrated significant advancements [10,11]. Machine learning techniques, particularly neural networks, have substantially improved accuracy and efficiency by capturing complex data patterns and interactions that traditional methods often overlook [12–14]. These capabilities have been effectively applied across various domains, including agricultural forecasting [15], commodity price prediction [16], carbon emissions pricing [17], and thermal coal futures trading volume prediction [18]. Similarly, machine learning has been leveraged to enhance SoC estimation. Various approaches, including supervised learning, neural networks, and ensemble methods, have been explored to predict battery SoC with high precision. For instance, the application of artificial neural networks (ANNs) and deep learning (DL) models has shown promising results in modeling battery behavior and improving SoC predictions. Research has highlighted the effectiveness of these models in capturing nonlinear relationships in battery data, leading to more accurate SoC estimates. Several studies [19,20] demonstrated how deep-learning models can enhance SoC estimation by incorporating historical data and battery characteristics. A related study [21] explored hybrid DL combined with metaheuristic optimization techniques for Nissan Leaf batteries, while Sulaiman et al. [22] employed an evolutionary mating algorithm (EMA) to optimize DL parameters. Early machine learning applications in SoC estimation included support vector regression (SVR), optimized using simulated annealing [23]. This model was integrated into a Kalman filter framework and combined with an ampere-hour integration method to create a closed-loop SoC estimation system. The model was trained under specific conditions and tested at various temperatures to evaluate its performance. A similar study [24], utilized SVR for state-of-health (SoH) estimation, predicting battery degradation and remaining useful life. This approach highlights the versatility of SVR in battery management applications, extending beyond SoC estimation to comprehensive health monitoring and prediction systems.

Additionally, support vector machines (SVMs) and other advanced machine learning techniques have been employed to refine SoC estimation further. The study by Korkmaz [10] investigated various machine learning algorithms, including bagging and extra trees, for SoC estimation, comparing their performance under different conditions and incorporating filters for outlier removal. These advancements underscore the growing role of machine learning in optimizing BMS and battery performance. Recent research has continued refining SoC estimation techniques. For instance, Li et al. [11] presented a hybrid machine learning framework for joint SoC and SoH estimation of LIBs, assisted by fiber sensor measurements. Another study integrated convolutional neural networks with Gaussian process regression, demonstrating that updating capacity estimation and incorporating advanced measurement techniques significantly enhance accuracy and reliability, reducing error and variability.

Other advancements include the use of recurrent neural networks (RNNs). Chen et al. [25] explored the application of gated recurrent neural networks (GRNNs) with Kalman filtering for SoC prediction. This study introduced the gated recurrent unit-adaptive Kalman filter (GRU-AKF), which effectively reduces SoC fluctuations and improves estimation accuracy across a wide temperature range. Another study [26] investigated a random search-optimized long short-term memory (RS-LSTM) network for SoC estimation. This approach highlights the effectiveness of the RS-LSTM approach in enhancing the estimation accuracy by optimizing key parameters and incorporating critical features, achieving a mean absolute error of 0.221% and a root mean square error of 0.262% across various conditions and datasets. Furthermore, Feng et al. [27] explored the integration of ensemble learning methods with feature selection techniques to enhance SoC prediction models, offering a robust framework for addressing battery performance complexities under varying operating conditions.

Previous studies on SoC estimation have explored various methods, including Coulomb counting, model-based approaches, and machine learning techniques. While Coulomb counting and model-based methods such as Kalman filters and electrochemical models are widely used, they suffer from limitations such as error accumulation and reliance on accurate battery models. Machine learning approaches, particularly ANNs, DL models, and hybrid methods incorporating optimization algorithms, have significantly improved estimation accuracy by capturing complex nonlinear relationships in battery data. Despite these advancements, challenges remain in optimizing hyperparameters, ensuring robustness across varying conditions, and reducing computational complexity. Recent developments have introduced techniques such as gated RNNs, long short-term memory (LSTM) networks, and ensemble learning methods, which enhance performance under diverse conditions. However, many of these methods require fine-tuning numerous parameters, making them computationally expensive.

To improve SoC predictions, selecting an effective algorithm is essential. In this study, deep neural networks (DNNs) were employed due to their ability to effectively model complex data relationships [28]. DNNs offer superior learning and adaptability compared to traditional machine learning models, particularly for time-series prediction and analysis [29]. To enhance DNNs performance, the teaching-learning based optimization (TLBO) algorithm was applied to optimize the network's weights and biases. TLBO is widely recognized for its straightforward implementation and effectiveness across various optimization tasks. Its user-friendly nature and minimal prior knowledge requirements render it practical for many applications [30]. Notably, it is free from algorithm-specific parameters [31], making it an attractive alternative to algorithms such as genetic algorithms (GA), which necessitate adjusting multiple parameters, including population size, number of generations, crossover probability, and mutation probability [32,33]. Moreover, TLBO exhibits strong convergence properties and delivers reliable solutions, making it a valuable tool for improving SoC prediction models [34].

The main contributions of the proposed TLBO-DNNs method are as follows:

i. By integrating DNNs with TLBO, the approach effectively addresses hyperparameter optimization challenges associated with traditional methods and reduces the risk of overfitting caused by minor variations in DNNs configurations.
ii. The TLBO-DNNs method outperformed established hybrid approaches, as demonstrated by experimental results on specific datasets, achieving significantly lower error rates compared with DNNs optimized by BMO, EMA, and particle swarm optimization (PSO).

The remainder of this paper is organized as follows: Section 2 presents the mathematical formulation of TLBO, followed by an overview of DNNs in Section 3. Section 4 details the methodology, including data collection, training, testing, the hybrid TLBO-DNNs model, and evaluation. Section 5 discusses the results, and Section 6 provides concluding remarks.

2. TBLO

The TLBO algorithm is inspired by the teaching and learning process in a classroom, as developed in [35,36]. Like other metaheuristic algorithms, TLBO follows a population-based approach, where the population is viewed as a group of learners, the design variables represent subjects, and the learners' performance corresponds to "fitness". The knowledge acquisition process is divided into two phases: the teacher phase and the learner phase, with the teacher being the best solution identified thus far.

This section describes the mathematical model of the TLBO algorithm, which consists of two main phases: teacher and learner.

2.1. Teacher phase

In this phase, Teacher T aims to improve the average performance of the class. The difference between the teacher's and students' average results for each subject is calculated as follows [35]:

$$Difference_{mean}^{j,k,i} = r_i \cdot \left(X_{j,kbest,i} - T_F M_{j,i}\right) \tag{1}$$

where $X_{j,kbest,i}$ represents the best learner's result in subject j during iteration i; k=1,2,…,n; $M_{j,i}$ denotes the mean result of the learners in a specific subject j (j=1,2,…, m); m refers to the number of design variables (i.e., subjects); T_F represents the teaching factor; r_i is a random number between 0 and 1 that influences the updates; and α and β (see Section 4) directly control the magnitude of the updates in the proposed method. T_F is calculated as follows [35]:

$$T_F = round[1 + rand(0,1)] \tag{2}$$

The performance of the TLBO improves when T_F is set to 1 or 2. Based on Eq. (1), the updated solution is defined as follows [35]:

$$X'_{j,k,i} = X_{j,k,i} + Difference_{mean}^{(j,k,i)} \tag{3}$$

where $X'_{j,k,i}$ represents the updated value of $X_{j,k,i}$, which is accepted if it improves the functional value. All accepted values serve as inputs for the learner phase.

2.2. Learner phase

In this phase, learners randomly interact with each other to enhance their knowledge. Given a population size of n, two learners, P and Q, are randomly selected such that $X'_{total-P,i} \neq X'_{total-Q,i}$, where $X'_{total-P,i}$ and $X'_{total-Q,i}$ are the updated values of $X_{total-P,i}$ and $X_{total-Q,i}$ of P and Q at the end of teacher phase, respectively [35]:

$$X''_{j,P,i} = X'_{j,P,i} + r_i\left(X'_{j,P,i} - X'_{j,Q,i}\right), If\ X'_{total-P,i} < X'_{total-Q,i} \tag{4}$$

$$X''_{j,P,i} = X'_{j,P,i} + r_i\left(X'_{j,Q,i} - X'_{j,P,i}\right), If\ X'_{total-Q,i} < X'_{total-P,i} \tag{5}$$

In the TLBO-DNNs hybrid model, the weights and biases of the DNNs correspond to $X_{j,k,i}$ whereas the teaching factor (*TF*) and random factor (r_i) control the magnitude of updates. Additionally, step sizes α and β are introduced to fine-tune the optimization for DNNs-specific applications.

3. Estimation based on DL

DL is a subset of machine learning that utilizes multi-layered neural networks, known as DNNs, to model complex patterns in data. These networks autonomously learn hierarchical features, making them highly effective for tasks such as image recognition, natural language processing, and predictive modeling. DNNs consist of multiple layers of interconnected neurons, where each neuron processes inputs using learned weights and activation functions to produce an output. The network's multi-layer architecture progressively extracts abstract features as the data pass through.

This study focuses on optimizing DNNs by determining the optimal weights and biases to enhance their performance. A novel approach hybrids DNNs with the TLBO to improve learning efficiency and accuracy. By optimizing the neural network parameters, this hybrid method aims to achieve better predictive performance, particularly for SoC estimation.

The forward pass in DNNs, which is essential for computing outputs based on inputs, weights, and biases, is defined as follows:

$$y = f\left(wx^T + b\right) \tag{6}$$

where y represents the output (SoC), w denotes the weights, x is the input data (see Section 4), b is the bias term, and f is the activation function.

4. Methodology

This section outlines the methodology used by TLBO-DNNs to estimate the SoCs of LIBs. It includes descriptions of the dataset, dataset analysis, data normalization, TLBO-DNNs, benchmarking techniques, and performance evaluation criteria. Fig. 1 illustrates the TLBO-DNNs framework for SoC estimation.

This study presents a hybrid approach that combines DNNs with a TLBO optimizer for SoC estimation in LIBs. It begins with a comprehensive dataset of battery parameters, including voltage, current, temperature, and various operational metrics. The data were divided into training (70%) and testing (30%) sets. DNNs were initially trained on these data to learn the complex relationships between the input features and SoC. The TLBO optimizer then iteratively refines the weights and biases of the DNNs, improving their ability to generalize and reducing overfitting. This optimization process continues until a termination criterion is met, resulting in the final TLBO-DNNs model. The model was evaluated using the testing set, with the output representing the estimated SoC percentage. This hybrid approach leverages the powerful learning capabilities of DNNs with the stabilizing effects of TLBO, offering potentially more accurate and robust SoC estimations than traditional methods or standalone DNNs. Details of each component are described in the following sections.

4.1. Dataset description

The quality of data significantly affects the performance of machine learning models; therefore, careful data collection and processing are crucial. In this study, data from 72 real driving trips of the BMW i3 (60 Ah) were used to validate a comprehensive vehicle model, which can be retrieved from [37]. The estimation model incorporated 10 input

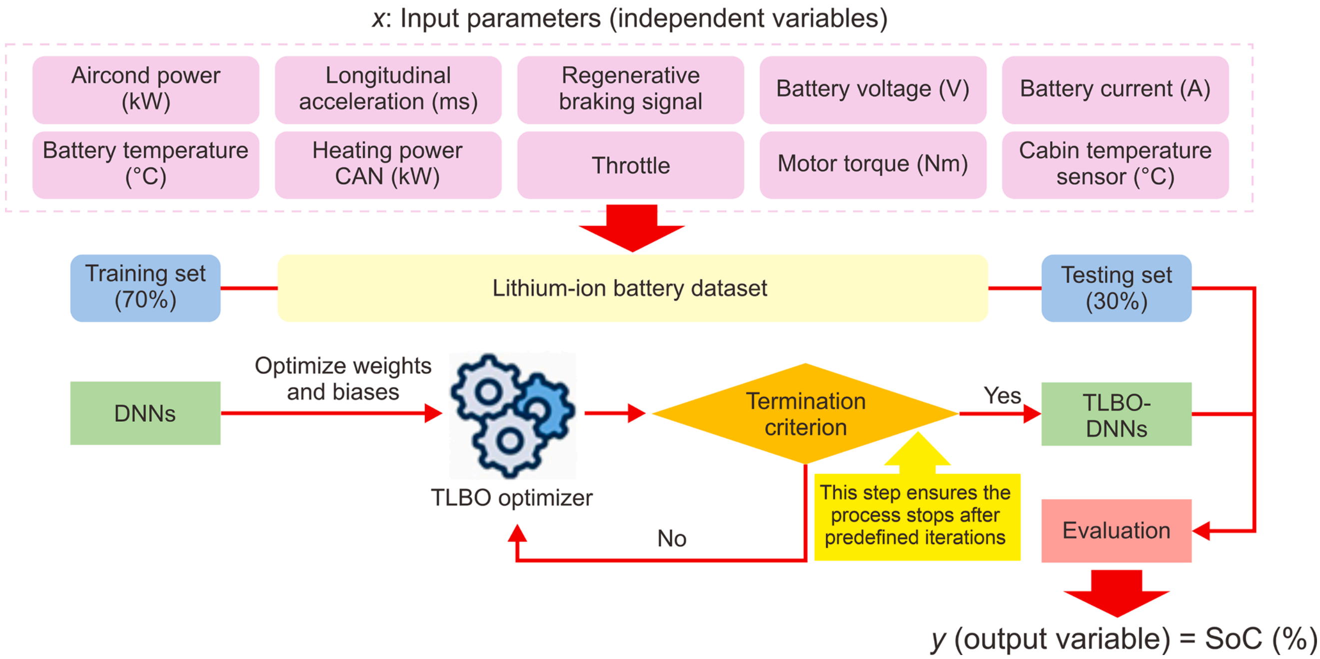

Fig. 1. Teaching-learning based optimization-deep neural networks (TLBO-DNNs) framework for state of charge (SoC) estimation.

parameters: air conditioning power (AC Power) (kW), longitudinal acceleration (LA) ($m{\cdot}s^{-2}$), regenerative braking signal (RBS), battery voltage (V_batt) (V), battery current (I_batt) (A), battery temperature (T_batt) (°C), heating power CAN (kW), throttle position (TP), motor torque (T_motor) (Nm), and cabin temperature (T_cabin). The model output was the SoC, expressed as a percentage. Table 1 lists the input samples.

4.2. Data normalization

To ensure consistent scaling and improve model performance, min–max normalization was applied to the dataset. This technique transforms the data into a specific range, typically [0, 1], helping maintain numerical stability and enhancing the convergence of optimization algorithms [38,39]. Min–max normalization was chosen for its simplicity and effectiveness in rescaling features to a uniform range. By compressing the data into a bounded interval, this method prevents large values from dominating the learning process and ensures that model parameters are updated uniformly. This normalization technique is particularly useful when features have varying scales or units, allowing for more stable and efficient model training. The min–max normalization formula is expressed as follows:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{7}$$

where x, min(x), max(x), and x' are the original, minimum, maximum, and normalized values of the feature, respectively.

4.3. TBLO-DNNs model

This methodology employs a hybrid approach that integrates TLBO with DNNs to optimize their performance. TLBO is used to automatically adjust the weights and biases of DNNs, enhancing their accuracy and efficiency. The application of TLBO for optimization is crucial as it enables effective exploration of the large and complex parameter space of deep neural networks. By automating the optimization process, TLBO efficiently identifies optimal or near-optimal values for weights and biases, a task that is otherwise challenging and time-consuming when performed manually. This approach not only improves DNNs performance but also ensures the model is fine-tuned to accommodate various data characteristics and complexities, leading to more accurate and robust predictions. The hybrid TLBO-DNNs framework is illustrated in Fig. 2.

Fig. 2 depicts a hybrid algorithm that hybrids TLBO with DNNs (TLBO-DNNs) for SoC estimation. The process begins with the initialization of TLBO-DNNs, followed by parameter configuration and the

Table 1
Sample input data.

AC power (kW)	LA ($m{\cdot}s^{-2}$)	RBS	V_batt (V)	I_batt (A)	T_batt (°C)	Heating power CAN (kW)	TP	T_motor (Nm)	T_cabin
0.4	−0.03	0	391.4	−2.2	21	0	0	0	24.5
0.4	0	0	391.4	−2.21	21	0	0	0	24.5
0.4	−0.01	0	391.4	−2.26	21	0	0	0	24.5
0.4	−0.03	0	391.4	−2.3	21	0	0	0	24.5
0.4	−0.03	0	391.4	−2.3	21	0	0	0	24.5
0.4	−0.01	0	391.4	−2.3	21	0	0	0	24.5
0.4	−0.01	0	391.4	−2.3	21	0	0	0	24.5
0.4	−0.03	0	391.4	−2.31	21	0	0	0	24.5
0.4	−0.01	0	391.4	−2.36	21	0	0	0.38	24.5
0.4	−0.01	0	391.4	−2.37	21	0	0	0.12	24.5

Note: AC power: air conditioning power; LA: longitudinal acceleration; RBS: regenerative braking signal; V_batt: battery voltage; I_batt: battery current; T_batt: battery temperature; TP: throttle position; T_motor: motor torque; T_cabin: cabin temperature.

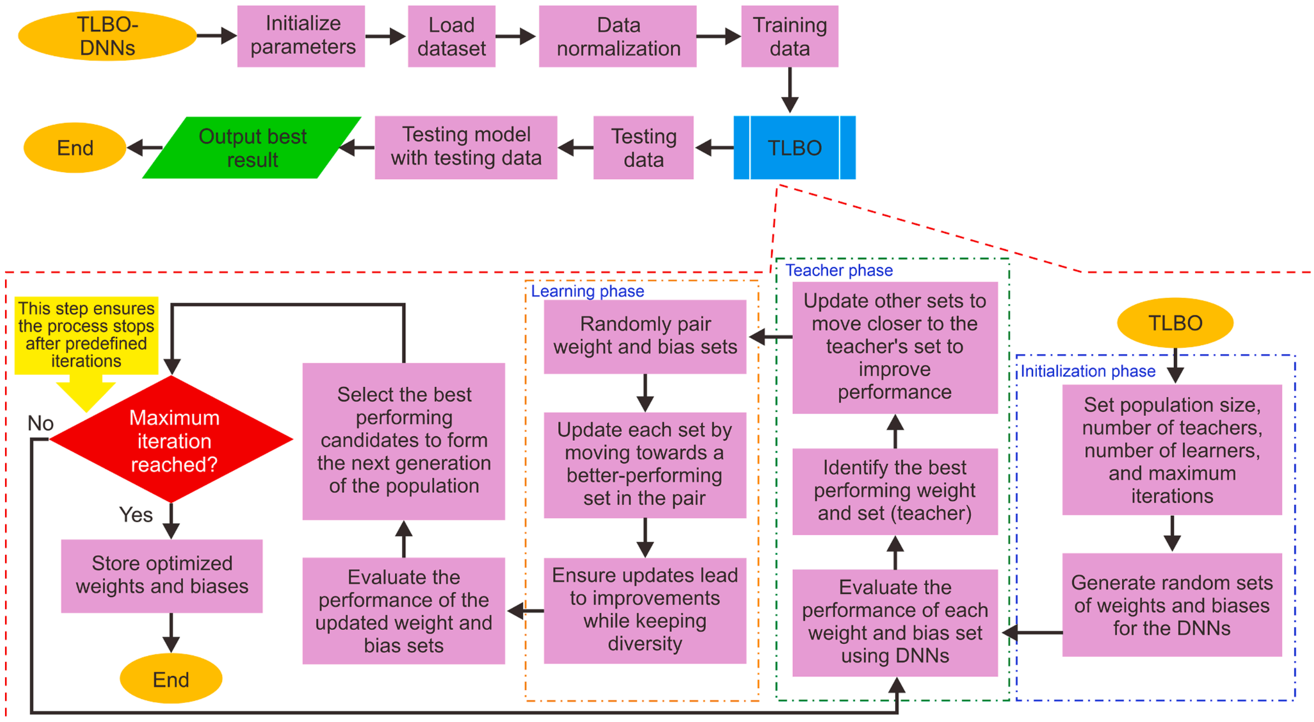

Fig. 2. Teaching-learning based optimization-deep neural networks (TLBO-DNNs) structure.

loading of driving-trip data from the BMW i3 dataset. After applying min–max normalization, the dataset is divided into training and testing sets. The TLBO process consists of two primary phases: initialization, teacher and learning.

4.3.1. Initialization phase

During the initialization phase, population parameters such as population size n, the number of teachers T, the number of learners L, and the maximum number of iterations $Iter_{max}$ are defined. In the context of DNNs, the population represents candidate sets of weights W and biases B, which correspond to the design variables in the TLBO framework. A random set of weights W_0 and biases B_0 is generated for the DNNs, where:

$$W_0 = rand(n, n_{weights}) \text{ and } B_0 = rand(n, n_{biases}) \tag{8}$$

4.3.2. Teacher phase

In the teacher phase, the best-performing candidate (referred to as the "teacher") is identified based on the DNN's performance. The weights and biases of the learners are updated to move closer to the teacher's solution. This process is governed by the teaching factor (TF), calculated as:

$$TF = round[1 + rand(0,1)] \tag{9}$$

The update equations for weights and biases are:

$$W^{new}_{\text{learner}} = W_{\text{learner}} + \alpha \cdot (W_{\text{teacher}} - W_{\text{learner}}) \tag{10}$$

$$B^{new}_{\text{learner}} = B_{\text{learner}} + \alpha \cdot (B_{\text{teacher}} - B_{\text{learner}}) \tag{11}$$

where α is the step size or learning rate, which controls the magnitude of the update. The learner's weights and biases are adjusted to enhance performance relative to the teacher.

4.3.3. Learning phase

In the learning phase, random pairs of weight and bias sets are formed, and each set is updated by moving toward the better-performing set within the pair. The update is expressed as:

$$W^{new}_i = W_i + \beta \cdot (W_{Best} - W_i) \tag{12}$$

$$B^{new}_i = B_i + \beta \cdot (B_{Best} - B_i) \tag{13}$$

where β is another step size that controls the magnitude of the update. This process iteratively refines the population by guiding the candidates toward improved solutions.

The optimization process continues iteratively, with model performance evaluated after each cycle. The iterations proceed until the maximum number of iterations $Iter_{max}$ is reached. At this stage, the TLBO-DNNs algorithm outputs the optimal weight and bias sets, refining the DNNs model to enhance SoC estimation accuracy.

This hybrid approach effectively combines the optimization capabilities of TLBO with the modeling strengths of DNNs, significantly improving the overall performance of the SoC estimation model.

4.4. Benchmarking techniques

This section provides a concise overview of the selected benchmarking techniques, which include three hybrid DNNs combined with metaheuristic optimization algorithms, namely, BMO, EMA, and PSO, as well as standalone models such as ARIMA and SVM.

4.4.1. BMO

The BMO [40] is a metaheuristic optimization algorithm inspired by the unique mating behavior of barnacles. This algorithm simulates the reproductive processes of barnacles, which rely on specific mating strategies and environmental adaptations. BMO models a population of candidate solutions that interact and "mate" based on their fitness levels, guiding the search for optimal solutions through mechanisms akin to genetic recombination and mutation. By leveraging these natural behaviors, BMO effectively explores and exploits the solution space, making it well-suited for solving complex optimization problems. It has been successfully applied to engineering design, scheduling, and other challenging tasks where traditional methods may be insufficient.

In BMO, new offspring are generated through a fertilization process involving a neighboring solution. Barnacles are known for their remarkably long penises—approximately seven times their body length—an adaptation that facilitates reproduction despite their sedentary lifestyle and changing tidal conditions. In BMO, the selection of barnacle parents for offspring generation is random, with the length of the barnacle's penis, denoted as *pl*, serving as a tuning parameter. The exploitation process is inspired by the Hardy-Weinberg principle, while exploration is guided by sperm casting mechanisms.

4.4.2. EMA

The EMA [41] is another metaheuristic optimization technique inspired by biological mating processes, particularly those described by Hardy-Weinberg (HW) principles [42]. Like other evolutionary algorithms, EMA consists of three fundamental stages: initialization, selection, and reproduction. However, it also incorporates environmental influences—specifically, the presence of predators—as an exploratory mechanism within its framework. Several key parameters influence the performance of EMA. These include population size, which determines the number of solutions per generation; the number of parents selected for mating, which affects genetic diversity; the mutation rate, which controls the frequency of random alterations in offspring; and the crossover rate, which dictates how often parent solutions combine to generate new offspring.

4.4.3. PSO

PSO [43] is a swarm-based metaheuristic algorithm in which a population of potential solutions, called particles, moves through the search space to find an optimal solution. Each particle represents a candidate solution to the optimization problem. In PSO, particles adjust their positions based on their own experiences and those of their neighbors, using both social and cognitive behaviors to explore and exploit the solution space efficiently. PSO is known for its rapid convergence and relatively few algorithmic parameters [44]. The key tuning parameters in PSO include the number of particles, which determines the exploration and convergence behavior of the algorithm; the cognitive coefficient ($c1$), also known as the personal learning coefficient, which influences how much a particle is attracted to its own best-known position; the social coefficient ($c2$), also referred to as the global learning coefficient, which determines the impact of the best-known position found by the entire swarm; the inertia weight (w), which controls the influence of a particle's previous velocity on its current velocity, balancing exploration and exploitation; and the velocity limits, which define the upper and lower bounds for particle movement to ensure effective search dynamics. These parameters are crucial for guiding the swarm towards optimal solutions and ensuring an effective search process.

4.4.4. ARIMA

ARIMA is a widely used statistical model for time series forecasting and analysis. It combines three components—autoregressive (AR), integrated (I), and moving average (MA)—to model and predict future values based on past data. The ARIMA model is governed by three key parameters: AutoRegressive Order p, which defines the number of lagged observations included in the model, determining how many past values are used to predict future values, with a higher value of p indicating that more past values are considered; differencing order, d, which represents the number of differences needed to make the time series stationary and is used to remove trends or seasonality, making the data more suitable for modeling; and moving average order, q, which specifies the number of lagged forecast errors included in the model, helping smooth the time series by averaging past forecast errors.

4.4.5. SVM

SVMs are supervised machine learning algorithms widely used for classification, regression, and outlier detection. SVMs work by finding the optimal hyperplane that separates data points of different classes in a high-dimensional space. For non-linear problems, SVMs employ kernel functions, such as polynomial or radial basis function (RBF) kernels, to map the input data into a higher-dimensional space where a linear separator can be applied. Due to their robustness and ability to handle high-dimensional data, SVMs are particularly effective in scenarios with limited datasets and clear separation margins.

4.5. Parameter setting

Before conducting the experiments, the properties of the proposed technique and the identified techniques were defined, as shown in Table 2. This table presents the key configuration settings, including population size, iteration limit, and neuron parameters for the TLBO, EMA, BMO, and PSO prediction techniques used in this study. For ARIMA, the values of p, d, and q were determined based on experimental findings, with ranges of 0–5, 0–1, and 0–5, respectively.

4.6. Model performance evaluation

Selecting an appropriate performance evaluation metric is crucial for validating experimental results. In this study, two statistical indices were used: mean absolute error (MAE) and root mean square error (RMSE). While RMSE assigns greater weight to larger estimation errors, MAE treats all errors equally. The formulas for MAE and RMSE are as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widetilde{y}_i)^2 \tag{14}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widetilde{y}_i)^2} \tag{15}$$

where y_i denotes the actual value, $\widetilde{y}$ is the predicted value, and N is the total number of samples in the test set. It is widely accepted that lower MAE and RMSE values indicate greater predictive accuracy.

5. Results and discussion

To evaluate the performance of TLBO-DNNs, a series of experiments was conducted to determine the most suitable number of neurons for the network. This study employs the widely used heuristic formula $2n+1$ to calculate the number of hidden neurons, where n represents the number of input neurons. This formula, inspired by the universal approximation theorem and related interpretations of Kolmogorov's work [45,46], offers a theoretical basis for representing continuous functions in a single hidden layer network. Based on this principle, the experiments tested configurations with 19, 21, and 23 neurons, corresponding to slight

Table 2
Parameters used for different SoC estimation optimization techniques.

Parameter	TLBO	EMA	BMO	PSO
Population size	30			
Maximum iterations	500			
Number of hidden layers	2			
Number of neurons	21			
Cr	-	0.85	-	-
R	-	< 0.35	-	-
Pl	-	-	21	-
c1	-	-	-	1.5
c2	-	-	-	1.5
W	-	-	-	0.7

Note: SoC: state of charge; TLBO: teaching-learning based optimization; EMA: evolutionary mating algorithm; BMO: barnacles mating optimizer; PSO: particle swarm optimization; *cr*: crossover probability; *r*: random number; *pl:* penis length; *c1*: cognitive coefficient; *c2*: social coefficient; *w*: inertia weight; - indicates that the parameter is not applicable or utilized.

variations in the calculated values. The primary goal of these experiments was to mitigate the risks of overfitting and underfitting. Overfitting can occur if the network has too many neurons, leading to excessive reliance on training data and poor generalization to unseen data. Conversely, underfitting may arise if the network contains too few neurons, preventing it from capturing the complexity of the data. By testing these configurations, this study aims to identify an optimal balance that maximizes network performance while ensuring effective generalization.

Table 3 presents a comparison of MAE and RMSE for different neuron configurations (19, 21, and 23) in the TLBO-DNNs for SoC estimation. The results indicate that increasing the number of neurons led to a decrease in MAE, with optimal performance achieved using 21 neurons. This configuration resulted in the lowest MAE (3.4480) and RMSE (4.6487), indicating superior prediction accuracy. In comparison, the 23-neuron configuration exhibited slightly higher MAE (3.8543) and RMSE (5.1796) values. The 19-neuron model performed the worst among the three, with the highest MAE (4.0500) and RMSE (5.3496), making it the least suitable for accurate SoC estimation. Based on these results, the experiment proceeded with the 21-neuron configuration, prioritizing overall prediction accuracy, which is crucial for reliable SoC estimation in BMS. From a practical standpoint, this finding suggests that designing a TLBO-DNNs model with 21 neurons strikes a balance between computational efficiency and prediction accuracy, making it a viable choice for deployment in BMS.

Table 4 presents the error metrics (MAE and RMSE) for the various models used in SoC estimation. The results indicate that TLBO-DNNs delivered the most accurate predictions, achieving the lowest MAE (3.4480) and RMSE (4.6487). This model demonstrated an optimal architecture with 21 neurons, as deviations in either direction (increasing or decreasing the number of neurons) led to reduced accuracy. This observation underscores the importance of careful hyperparameter tuning for optimal performance. Compared with other neural network-based models, TLBO-DNNs consistently outperformed BMO-DNNs, EMA-DNNs, and PSO-DNNs. For instance, BMO-DNNs achieved an MAE of 5.3848 and RMSE of 7.0395, while PSO-DNNs performed slightly better than BMO-DNNs, with an MAE of 4.3089 and RMSE of 5.9672. In contrast, EMA-DNNs exhibited the highest error rates among neural network-based approaches, with an MAE of 7.6127 and RMSE of 11.2287, indicating slower convergence and less effective optimization.

ARIMA, a traditional statistical method, performed significantly worse, with an MAE of 14.3301 and RMSE of 17.0697. This stark contrast highlights the advantages of DL models in capturing the nonlinear relationships inherent in battery data. The superior performance of TLBO-DNNs is particularly noteworthy, as they consistently achieved the lowest errors across all tested models and evaluation metrics.

The results presented in Table 4 not only confirm the effectiveness of TLBO-DNNs but also highlight the limitations of ARIMA. While ARIMA serves as a useful benchmark for linear modeling, it struggles to capture the complex and dynamic patterns observed in BMS. To further validate these findings, future work should explore additional datasets, conduct statistical validations, and consider other performance metrics, such as execution time and energy efficiency. Nevertheless, the current results establish TLBO-DNNs as a highly promising solution for accurate SoC estimation in BMS, with their optimized architecture and robust learning capabilities outperforming both neural-network-based competitors and traditional methods.

Table 3
SoC estimation using different TLBO-DNNs model configurations with 19, 21, and 23 neurons.

Number of neurons	MAE	RMSE
19	4.0500	5.3496
21	3.4480	4.6487
23	3.8543	5.1796

Note: SoC: state of charge; TLBO-DNNs: teaching-learning based optimization-deep neural networks; MAE: mean absolute error; RMSE: root mean square error.

Table 4
SoC estimation using DNNs optimized through metaheuristic algorithms and ARIMA.

Algorithms	MAE	RMSE
TLBO-DNNs	**3.4480**	**4.6487**
BMO-DNNs	5.3848	7.0395
EMA-DNNs	7.6127	11.2287
PSO-DNNs	4.3089	5.9672
ARIMA	14.3301	17.0697
SVM	6.0065	8.0360

Note: SoC: state of charge; DNNs: deep neural networks; TLBO: teaching-learning based optimization; BMO: barnacles mating optimizer; EMA: evolutionary mating algorithm; PSO: particle swarm optimization; ARIMA: autoregressive integrated moving average; SVM: support vector machine; MAE: mean absolute error; RMSE: root mean square error.

For TLBO-DNNs, the results reveal a delicate balance between underfitting and overfitting. The model with 19 neurons exhibited slightly higher error rates than the 21-neuron model, suggesting mild underfitting, indicating that the 19-neuron architecture may lack the capacity to fully capture the complexity of SoC estimation. The optimal performance achieved with 21 neurons suggests that this configuration strikes a balance, effectively modeling the underlying patterns without overfitting to noise in the training data. However, the performance degradation observed when the number of neurons increased to 23 is a classic sign of overfitting. The additional complexity allowed the model to fit too closely to the training data, reducing its ability to generalize to new, unseen data.

Regarding the comparison across different models, the consistently superior performance of TLBO-DNNs suggests that they better capture the true underlying patterns of SoC behavior than other approaches. The progressively worse performances of BMO-DNNs, EMA-DNNs, and PSO-DNNs may indicate varying degrees of underfitting, with these models potentially lacking the necessary complexity or optimization strategies to fully capture SoC dynamics. The significantly poorer performance of ARIMA strongly suggests underfitting, likely due to its linear nature, which is unable to represent the nonlinear relationships in battery behavior. Figs. 3–7 illustrate the performance of TLBO-DNNs and other models.

Fig. 3 presents a visual comparison of actual and predicted energy values, along with the associated prediction errors for SoC estimation using TLBO-DNNs across approximately 350,000 instances. The top plot compares actual energy values (blue curve) with predicted values (orange curve). The close alignment between these curves highlights the effectiveness of TLBO-DNNs in accurately capturing SoC variations across the dataset. The small discrepancies between the two curves indicate that the model performed consistently well, even in regions with sudden fluctuations or nonlinear energy trends. The bottom plot shows the error percentages, revealing that most errors were centered around zero, with the majority falling within a narrow range, further demonstrating the robustness of TLBO-DNNs. Although occasional spikes in error are observed at specific instances, they remain minimal and do not significantly impact overall prediction accuracy. The insights from Fig. 3 confirm that TLBO-DNNs deliver reliable SoC estimation while maintaining high accuracy across a large dataset. The model's ability to handle both steady-state and transient energy variations further supports its suitability for real-world BMS.

In contrast, the error plot for BMO-DNNs (Fig. 4) exhibits significant fluctuations, with errors frequently exceeding ±10% and occasionally reaching ±40%, indicating substantial mispredictions. Even larger estimation errors are observed in Fig. 5, which illustrates the performance of EMA-DNNs.

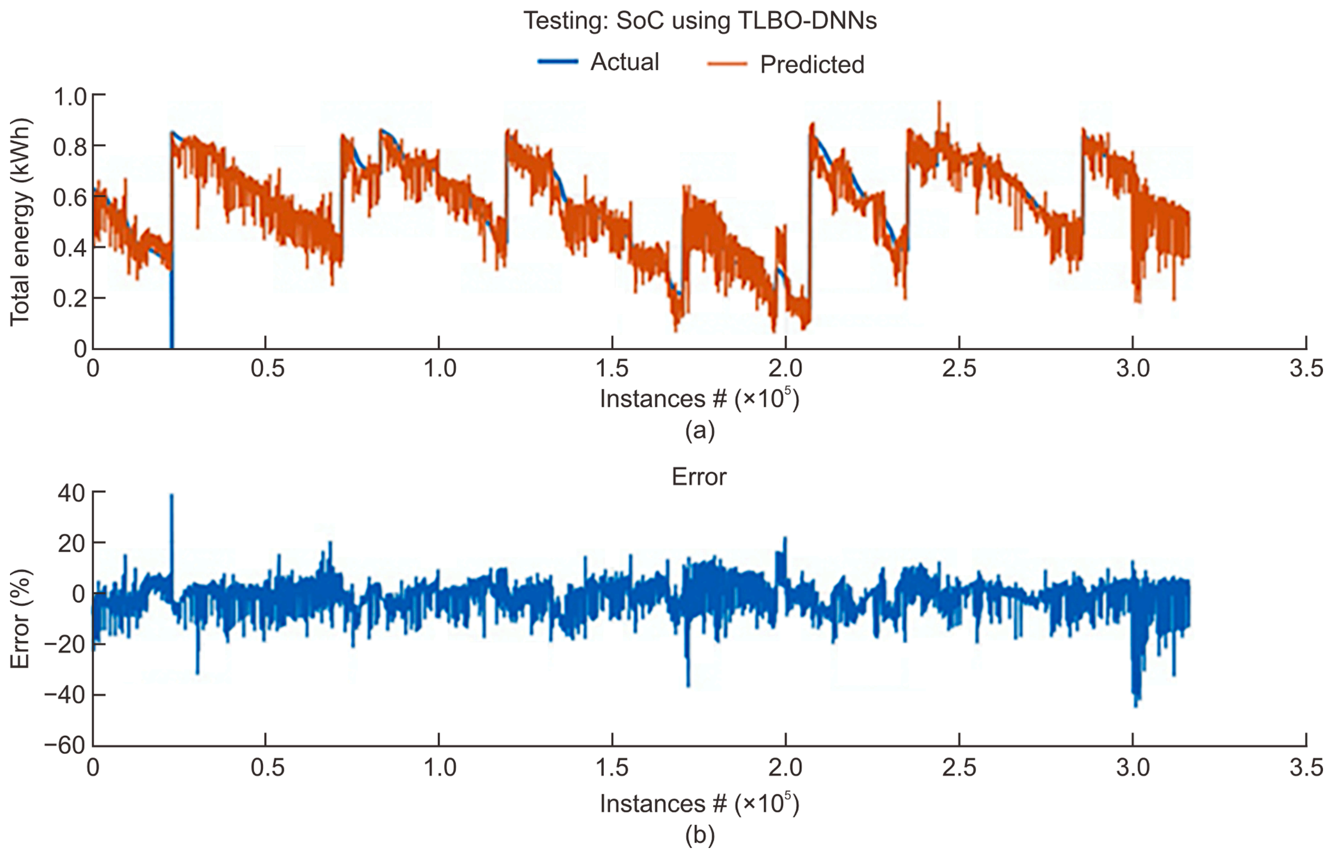

Fig. 3. State of charge (SoC) estimation using teaching-learning based optimization-deep neural networks (TLBO-DNNs).

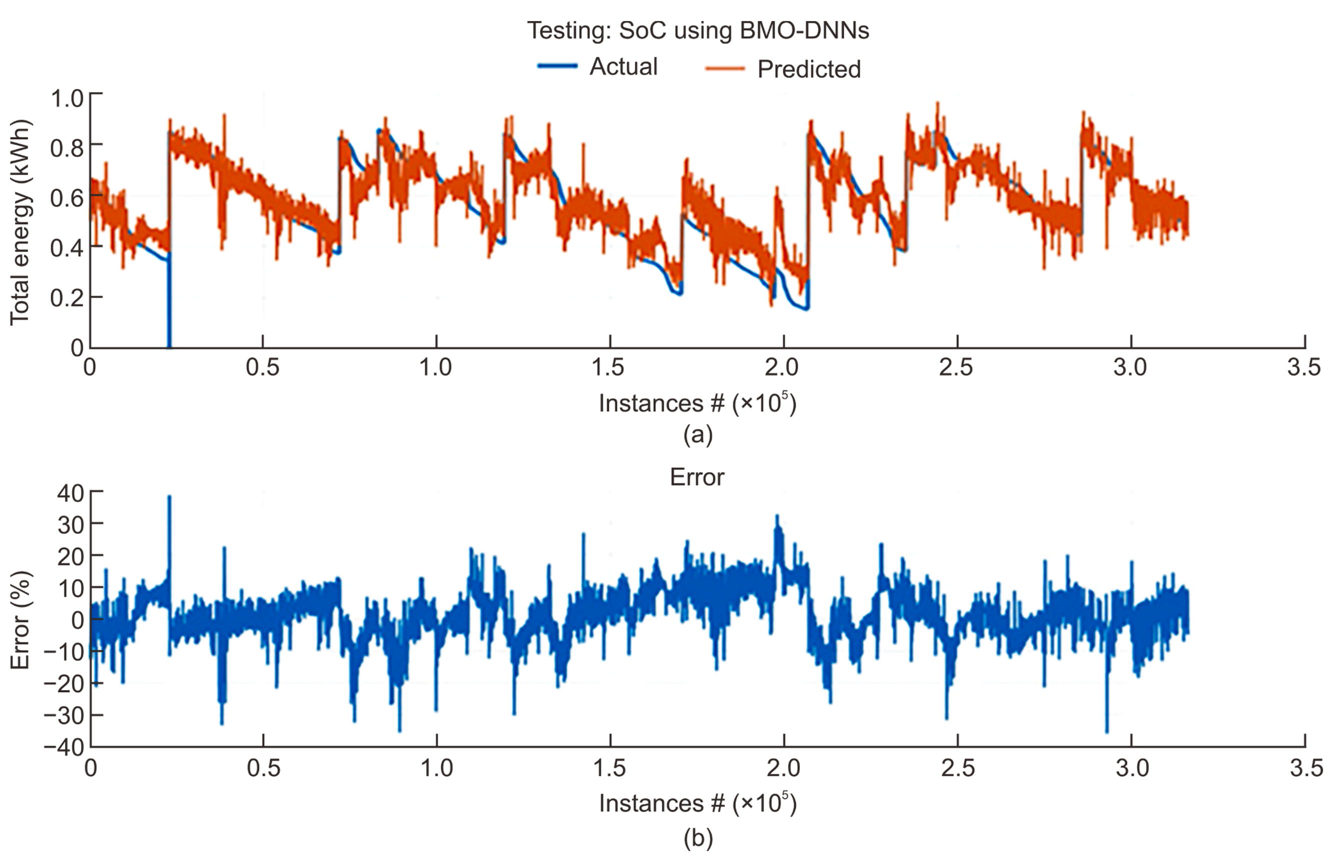

Fig. 4. State of charge (SoC) estimation using barnacle mating optimizer–deep neural networks (BMO-DNNs).

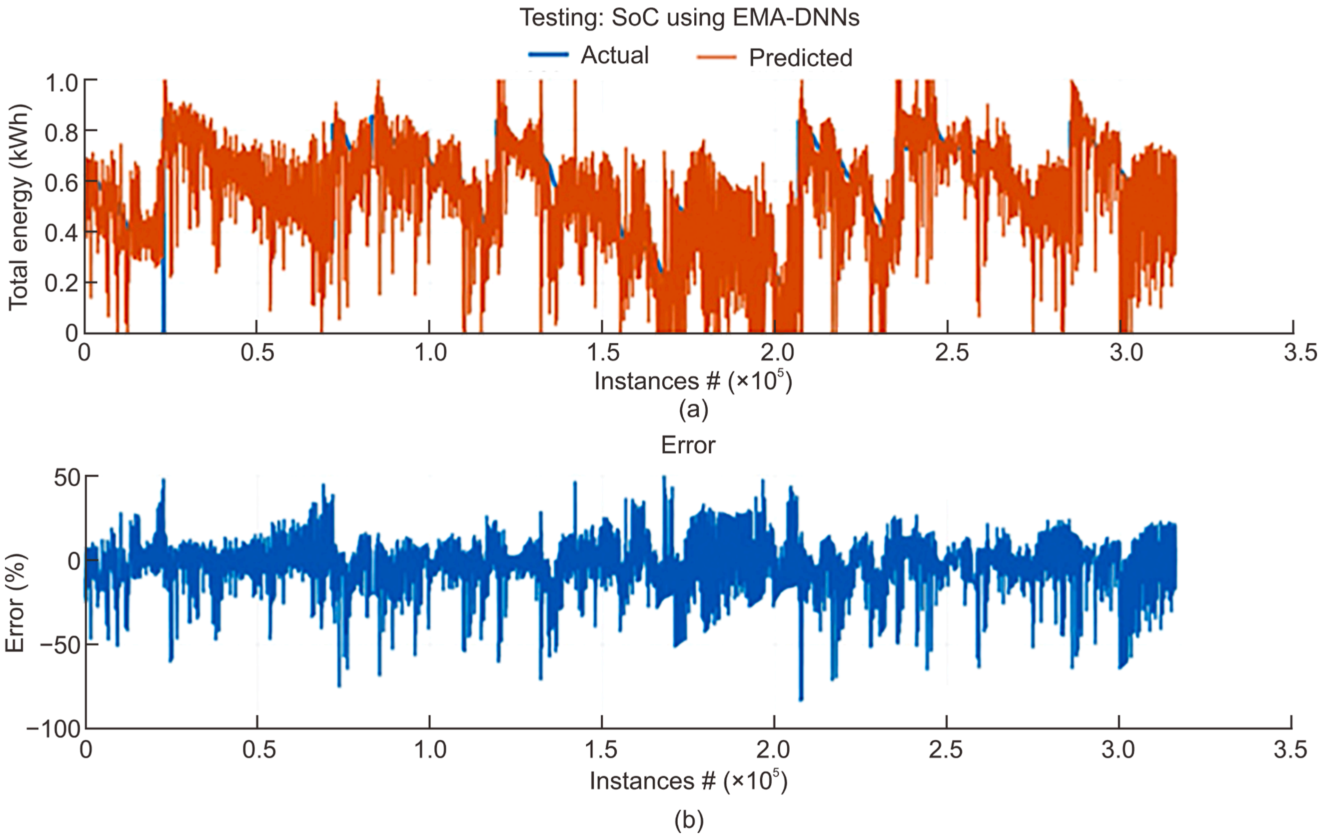

Fig. 5. State of charge (SoC) estimation using evolutionary mating algorithm-deep neural networks (EMA-DNNs).

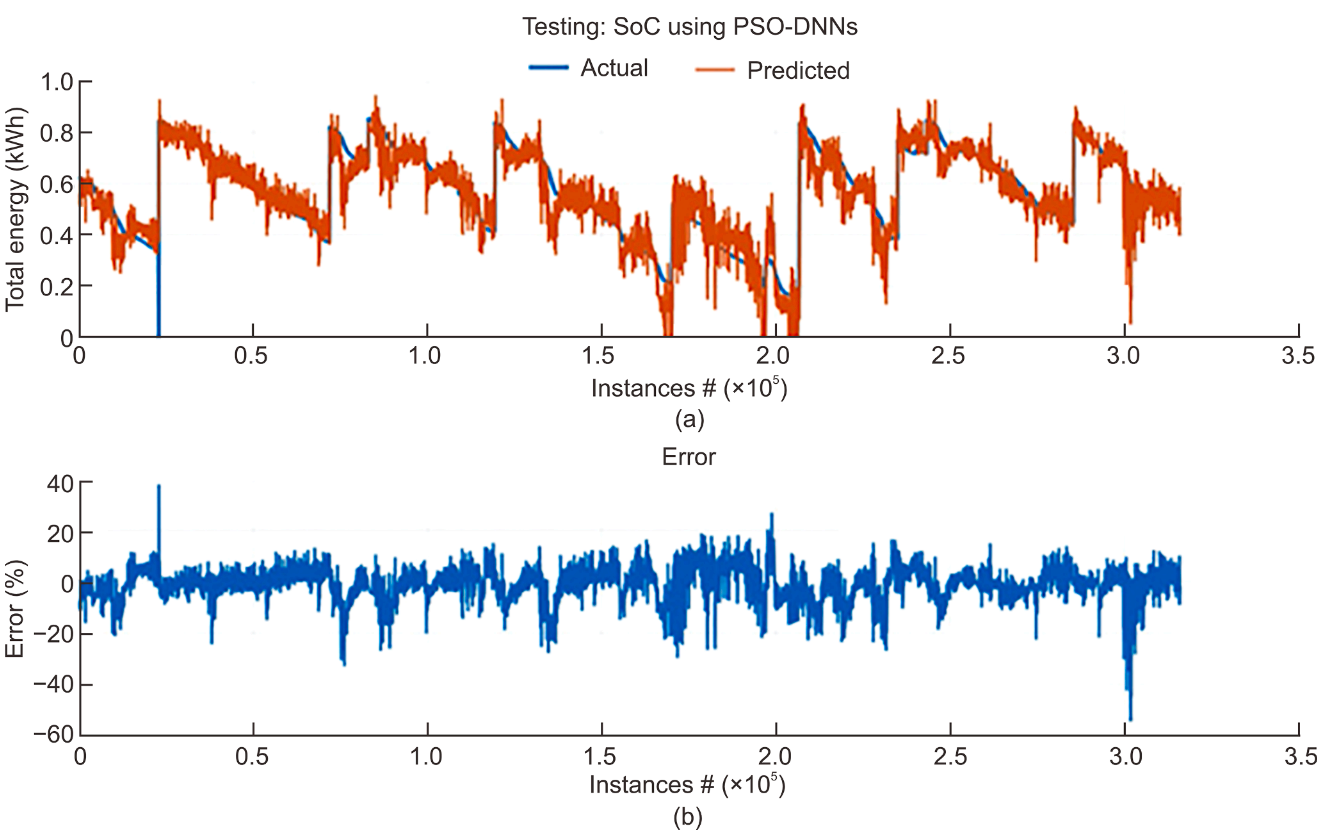

Fig. 6. State of charge (SoC) estimation using particle swarm optimization–deep neural networks (PSO-DNNs).

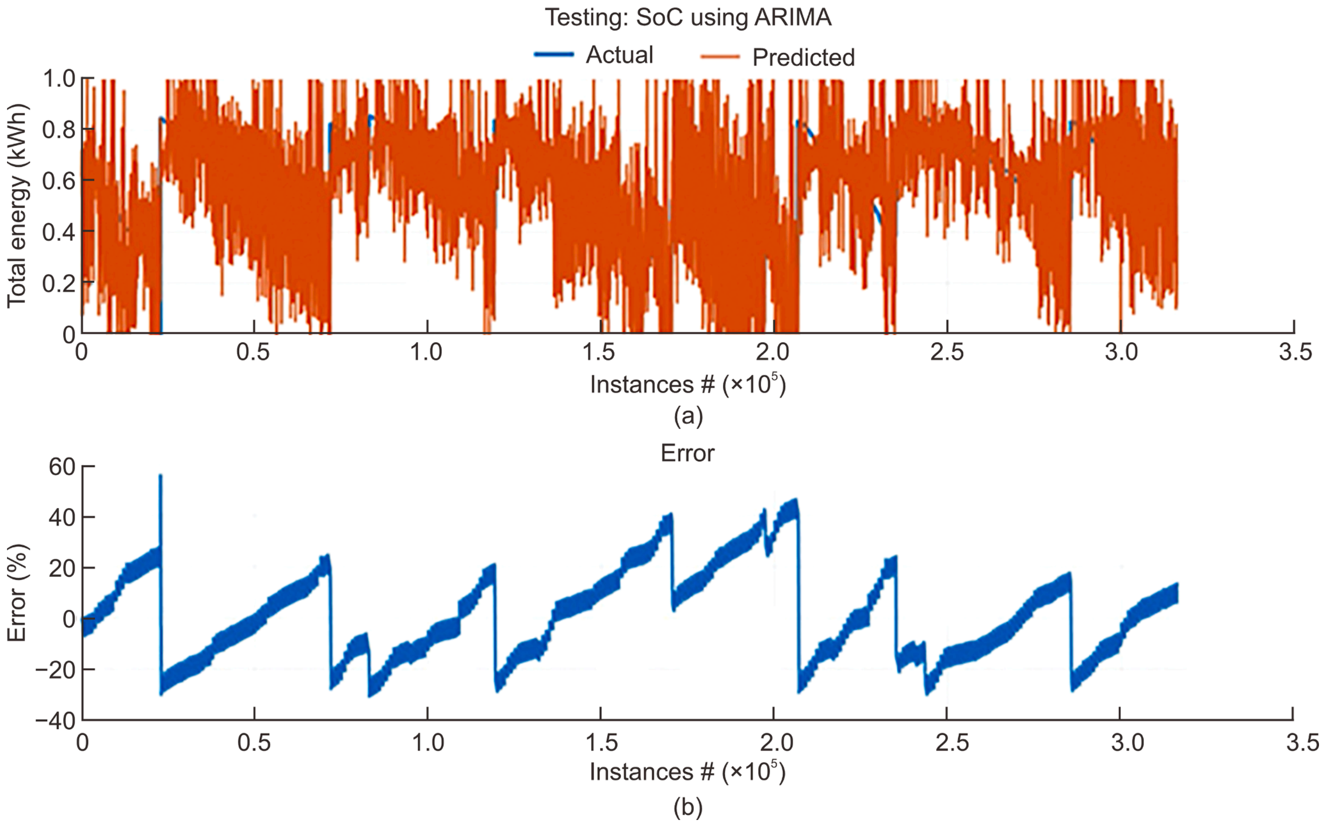

Fig. 7. State of charge (SoC) estimation using autoregressive integrated moving average (ARIMA).

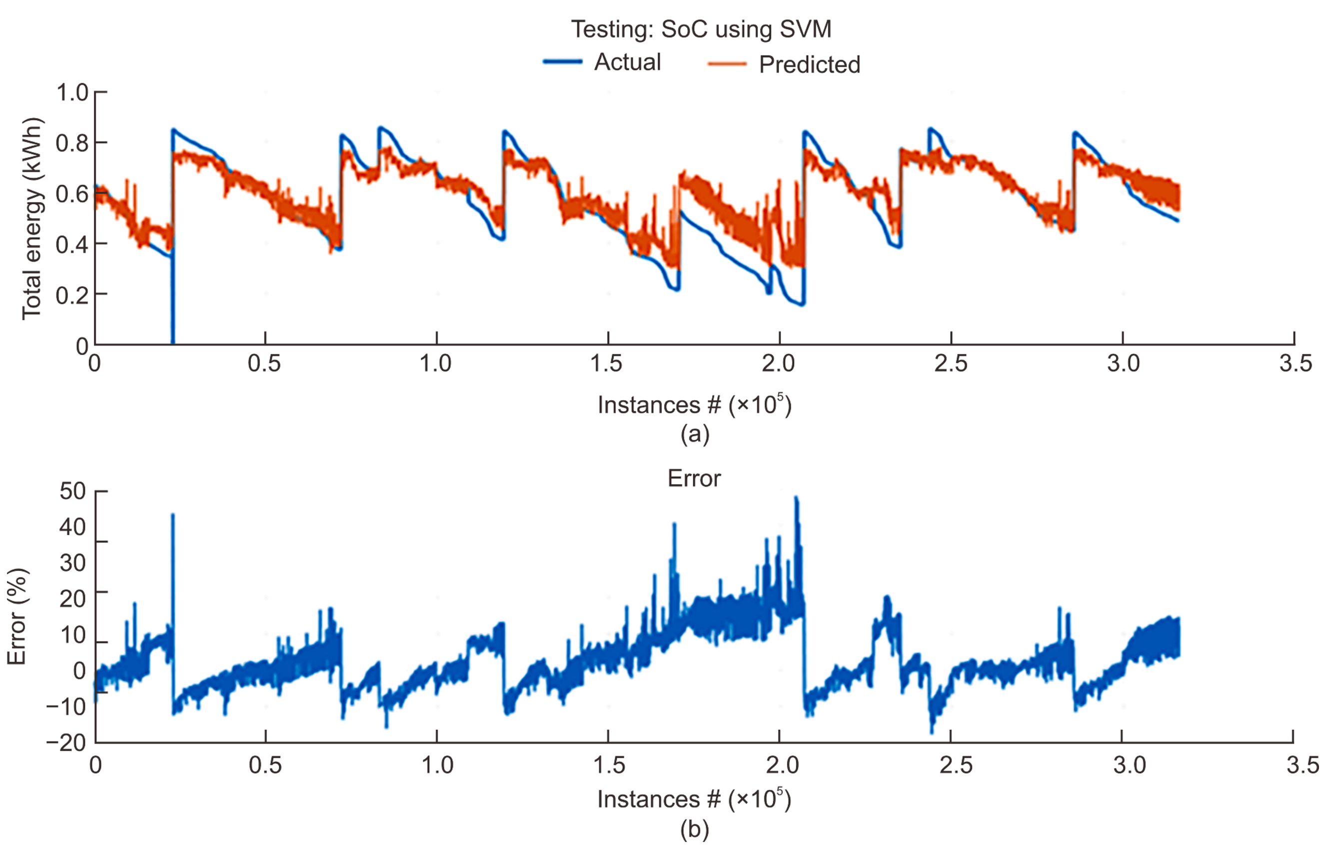

Fig. 8. State of charge (SoC) estimation using support vector machine (SVM).

Fig. 6 visualizes the SoC estimation using PSO-DNNs, which demonstrates better accuracy than BMO-DNNs. However, it exhibits frequent error spikes exceeding ±20% and occasionally approaching ±40%, in contrast to the more stable performance of TLBO-DNNs. Meanwhile, ARIMA showed poor performance for SoC estimation (see Fig. 7), with large errors, unstable predictions, and a clear inability to track actual energy levels accurately. These results reinforce the notion that ARIMA's limited adaptability and difficulty in capturing nonlinear relationships within the data diminish its effectiveness [47]. SVM exhibited moderate performance in predicting SoC, as shown in Fig. 8. While it outperformed ARIMA, it fell short compared to hybrid DNN approaches such as TLBO-DNNs and PSO-DNNs, which achieved significantly lower error values.

Fig. 9 presents the convergence curves of the four optimization algorithms—TLBO, EMA, PSO, and BMO—applied to SoC estimation. Among these, TLBO demonstrated the best performance, achieving the lowest final objective function value of 0.0328 after approximately 450 iterations, indicating its superior ability to minimize the objective function. PSO followed a similar convergence pattern but reached a slightly higher final value of 0.0345, demonstrating competitive performance. In contrast, BMO and EMA converged more slowly to higher objective function values of 0.0492 and 0.0536, respectively. The figure also shows that TLBO consistently outperformed the other algorithms throughout the iterations, maintaining lower objective function values at nearly all points. This superior performance, combined with its rapid and steady convergence, suggests that TLBO is the most suitable algorithm for this SoC estimation task. While PSO is a close competitor, BMO and EMA are less optimal due to their slower and less effective convergence behavior.

To further validate these observations and ensure robust model selection, future work should incorporate cross-validation techniques, analyze learning curves, and evaluate the models on separate validation and test sets. This would provide a more comprehensive understanding of each model's generalization capabilities and help fine-tune their architectures to achieve an optimal balance between complexity and performance.

6. Conclusion

This study introduces an innovative hybrid TLBO-DNNs approach for estimating the SoC of LIBs in BMW EVs. The TLBO algorithm was used to optimize the weights and biases of the DNNs, thereby improving SoC estimation accuracy. To ensure effective training, the dataset was normalized using min-max scaling to address variations in value ranges. Experiments were conducted with different numbers of hidden neurons to prevent underfitting or overfitting. The TLBO-DNN method was evaluated using a dataset of 1,064,000 samples and achieved an MAE of 0.34480 and an RMSE of 4.6487. Compared to other hybrid methods, such as BMO-DNNs, EMA-DNNs, and PSO-DNNs, as well as standalone models like ARIMA and SVM, TLBO-DNNs demonstrated superior performance. This hybrid approach has significant potential for enhancing BMS in EVs, leading to more efficient and reliable operation.

Despite its strong performance, TLBO-DNNs exhibited some error spikes in specific instances, suggesting that the model may struggle under certain conditions. These deviations could result from variations in battery behavior or limitations in the model's generalizability. Additionally, extreme conditions may challenge its predictive accuracy. To address these limitations, future work could focus on refining the model by incorporating more comprehensive features, exploring alternative data preprocessing techniques to better handle outliers, and improving its ability to generalize across diverse scenarios. Another avenue for improvement involves exploring advanced hybrid models or integrating domain-specific knowledge to enhance prediction accuracy in challenging situations. Strengthening these aspects would help ensure robust performance across a wider range of conditions, contributing to more accurate and reliable SoC estimation.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve language and readability, with caution. After using this tool/service, the authors reviewed and edited the content as needed and

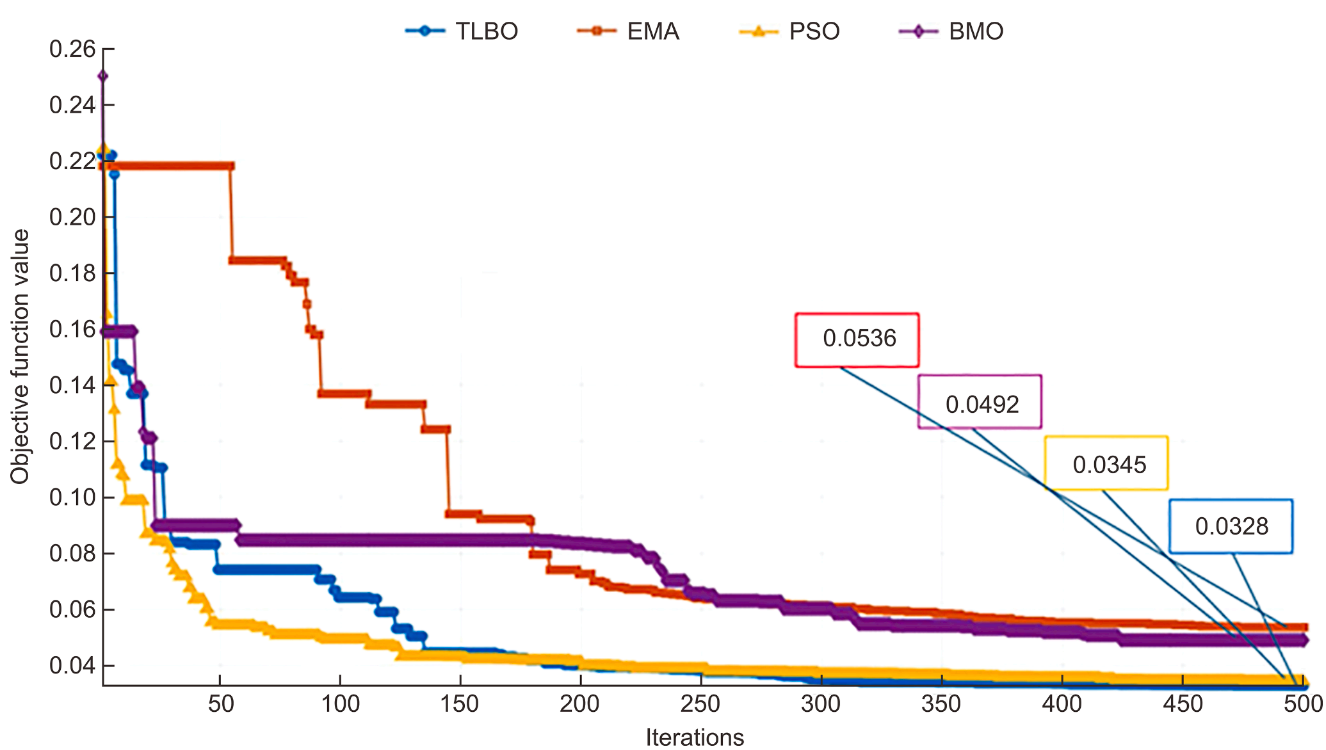

Fig. 9. Convergence curves obtained with teaching-learning based optimization (TLBO), evolutionary mating algorithm (EMA), particle swarm optimization (PSO), and barnacle mating optimizer (BMO).

take full responsibility for the content of the publication.

CRediT authorship contribution statement

Zuriani Mustaffa: Writing – original draft, Methodology, Data curation, Conceptualization. **Mohd Herwan Sulaiman:** Writing – review & editing, Validation, Formal analysis. **Jeremiah Isuwa:** Writing – review & editing.

Declaration of competing interest

The authors declare that there are no conflicts of interest.

Acknowledgements

This work was supported by Ministry of Higher Education Malaysia (MOHE) under Fundamental Research Grant Scheme (Grant No.: FRGS/1/2024/ICT02/UMP/02/).

References

[1] J. Maldonado, A. Jain, S. Castellanos, Assessing the impact of electric vehicles in Mexico's electricity sector and supporting policies, Energy Policy 191 (2024) 114152.
[2] J. Heo, S. Chang, Optimal planning for electric vehicle fast charging stations placements in a city scale using an advantage actor-critic deep reinforcement learning and geospatial analysis, Sustain. Citi. Soc. 113 (2024) 105567.
[3] M. Waseem, G.S. Lakshmi, E. Sreeshobha, et al., An electric vehicle battery and management techniques: comprehensive review of important obstacles, new advancements, and recommendations, Energy Storage Sav. 4 (2024) 83–108.
[4] H.A. Sakr, M.M. Fouda, A.F. Ashour, et al., Machine learning-based detection of DDoS attacks on IoT devices in multi-energy systems, Egypt. Inform. J. 28 (2024) 100540.
[5] Z.M. Ali, M. Calasan, F.H. Gandoman, et al., Review of batteries reliability in electric vehicle and E-mobility applications, Ain Shams Eng. J. 15 (2024) 102442.
[6] K. Das, R. Kumar, Electric vehicle battery capacity degradation and health estimation using machine-learning techniques: a review, Clean Energy 7 (2023) 1268–1281.
[7] H.M. Fahmy, H.M. Hasanien, I. Alsaleh, et al., State of health estimation of lithium-ion battery using dual adaptive unscented Kalman filter and coulomb counting approach, J. Energy Storage 88 (2024) 111557.
[8] Z.H. Zhou, M.J. Zhan, B.G. Wu, et al., A novel adaptive unscented Kalman filter algorithm for SOC estimation to reduce the sensitivity of attenuation coefficient, Energy 307 (2024) 132598.
[9] Q. Zhang, G.W. Wan, C.R. Li, et al., State of charge estimation for Li-ion battery during dynamic driving process based on dual-channel deep learning methods and conditional judgement, Energy 294 (2024) 130948.
[10] M. Korkmaz, SoC estimation of lithium-ion batteries based on machine learning techniques: a filtered approach, J. Energy Storage 72 (2023) 108268.
[11] Y.H. Li, K. Li, X. Liu, et al., A hybrid machine learning framework for joint SOC and SOH estimation of lithium-ion batteries assisted with fiber sensor measurements, Appl. Energy 325 (2022) 119787.
[12] B. Jin, X. Xu, Forecasting wholesale prices of yellow corn through the Gaussian process regression, Neural Comput. Appl. 36 (2024) 8693–8710.
[13] B.Z. Jin, X.J. Xu, Predictions of steel price indices through machine learning for the regional northeast Chinese market, Neural Comput. Appl. 36 (2024) 20863–20882.
[14] B.Z. Jin, X.J. Xu, Pre-owned housing price index forecasts using gaussian process regressions, J. Model. Manag. 19 (2024) 1927–1958.
[15] B.Z. Jin, X.J. Xu, Wholesale price forecasts of green grams using the neural network, Asia, J. Econ. Bank. (2024), https://doi.org/10.1108/AJEB-01-2024-0007.
[16] B.Z. Jin, X.J. Xu, Price forecasting through neural networks for crude oil, heating oil, and natural gas, Meas. Energy 1 (2024) 100001.
[17] B.Z. Jin, X.J. Xu, Carbon emission allowance price forecasting for China Guangdong carbon emission exchange via the neural network, Glob. Finance Rev. 6 (2023) 3491.
[18] B.Z. Jin, X.J. Xu, Y. Zhang, Thermal coal futures trading volume predictions through the neural network, J. Model. Manag. (2024), https://doi.org/10.1108/JM2-09-2023-0207.
[19] M.H. Sulaiman, Z. Mustaffa, S. Razali, et al., Advancing battery state of charge estimation in electric vehicles through deep learning: a comprehensive study using real-world driving data, Clean. Energy Syst. 8 (2024) 100131.
[20] J.P. Tian, C. Chen, W.X. Shen, et al., Deep learning framework for lithium-ion battery state of charge estimation: recent advances and future perspectives, Energy Storage Mater 61 (2023) 102883.
[21] 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.
[22] M.H. Sulaiman, Z. Mustaffa, N.F. Zakaria, et al., Using the evolutionary mating algorithm for optimizing deep learning parameters for battery state of charge estimation of electric vehicle, Energy 279 (2023) 128094.
[23] Y. Li, M. Ye, Q. Wang, et al., An improved model combining machine learning and Kalman filtering architecture for state of charge estimation of lithium-ion batteries, Green Energy Intell. Transp. 3 (2024) 100163.
[24] Y.F. Guo, K. Huang, X.Y. Yu, et al., State-of-health estimation for lithium-ion batteries based on historical dependency of charging data and ensemble SVR, Electrochim. Acta 428 (2022) 140940.
[25] J.X. Chen, Y. Zhang, W.J. Li, et al., State of charge estimation for lithium-ion batteries using gated recurrent unit recurrent neural network and adaptive Kalman filter, J. Energy Storage 55 (2022) 105396.
[26] X.Q. Chai, S.H. Li, F.W. Liang, A novel battery SOC estimation method based on random search optimized LSTM neural network, Energy 306 (2024) 132583.
[27] J.Q. Feng, F. Cai, Y. Zhao, et al., A novel feature optimization and ensemble learning method for state-of-health prediction of mining lithium-ion batteries, Energy 299 (2024) 131474.
[28] C.J. Lin, Z.H. Liu, M. Zhang, et al., Improving soil organic carbon estimation in paddy fields using data augmentation algorithm and deep neural network model based on optimal image date, Comput. Electron. Agric. 220 (2024) 108921.
[29] Y.P. Song, J.F. Huang, Y. Xu, et al., Multi-decomposition in deep learning models for futures price prediction, Expert Syst. Appl. 246 (2024) 123171.
[30] Y. Xia, F.X. Hu, H.J. Cao, et al., A blind color image watermarking algorithm based on Hadamard transform and TLBO algorithm, Optik 290 (2023) 171277.
[31] H.S. Gill, B.S. Khehra, A. Singh, et al., Teaching-learning-based optimization algorithm to minimize cross entropy for selecting multilevel threshold values, Egypt, Inform. J. (2022).
[32] S. Vitayasak, P. Pongcharoen, Performance improvement of Teaching-learning-based optimisation for robust machine layout design, Expert Syst. Appl. 98 (2018) 129–152.
[33] W.L. Li, H. Li, Y.T. Wang, et al., Optimizing flexible job shop scheduling with automated guided vehicles using a multi-strategy-driven genetic algorithm, Egypt, Inform. J. 25 (2024) 100437.
[34] M.K. Sun, Z.Y. Cai, H.N. Zhang, A teaching-learning-based optimization with feedback for L-R fuzzy flexible assembly job shop scheduling problem with batch splitting, Expert Syst. Appl. 224 (2023) 120043.
[35] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aid. Des. 43 (2011) 303–315.
[36] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems, Inf. Sci. Int. J. 183 20121–15.
[37] G. Wadhwa. Predicting car battery heating data. https://www.kaggle.com/code/gitanjali1425/predicting-car-battery-heating-data (Accessed 1 August 2024).
[38] Y.S. Kim, M.K. Kim, N.D. Fu, et al., Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models, Sustain. Citi. Soc. 118 (2025) 105570.
[39] L. Al Shalabi, Z. Shaaban, Normalization as a preprocessing engine for data mining and the approach of preference matrix, in: 2006 International Conference on Dependability of Computer Systems. May 25–27, 2006, IEEE, Szklarska Poreba, Poland, 2006, pp. 207–214.
[40] M.H. Sulaiman, Z. Mustaffa, M.M. Saari, et al., Barnacles Mating Optimizer: a new bio-inspired algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 87 (2020) 103330.
[41] M.H. Sulaiman, Z. Mustaffa, M.M.S. Saari, et al., Evolutionary mating algorithm, Neural Comp. Appl. 35 (2022) 487–516.
[42] J. Lachance, Hardy–Weinberg equilibrium and random mating. Encyclopedia of Evolutionary Biology, Elsevier, Amsterdam, 2016, pp. 8–211.
[43] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks. November 27 - December 1, 1995, Perth, Australia, IEEE, 2002, pp. 1942–1948.
[44] X. Ju, S.K. Liu, Y.L. Xiao, et al., Prediction model for cost data of a power transmission and transformation project based on Pearson correlation coefficient–IPSO–ELM, Clean. Energy 5 (2021) 756–764.
[45] R. Hecht-Nielsen, Kolmogorov's mapping neural network existence theorem. Proceedings of the International Conference on Neural Networks 3, 1987, pp. 11–14.
[46] S. Karsoliya, M. Azad, Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture 3 (2012) 714–717.
[47] X.K. Wu, X.Y. Gu, K.W. See, ADNNet: attention-based deep neural network for Air Quality Index prediction, Expert. Syst. Appl. 258 (2024) 125128.