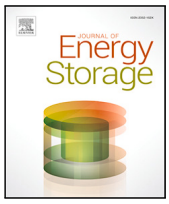
Journal of Energy Storage 119 (2025) 115948

Contents lists available at ScienceDirect

Journal of Energy Storage

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



Research papers

A hybrid model with Gaussian process-based covariance matrix adaptation evolution strategy for state of charge estimation of lithium-ion batteries

Wei Wang [a], Yong Wang [a,*], Zijun Zhang [b]

[a] School of Automation, Central South University, Changsha 410083, China
[b] Department of Data Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong SAR, China

ARTICLE INFO

Keywords:
Lithium-ion battery
State of charge
Covariance matrix adaptation evolution strategy
Gate recurrent unit network
Kalman filtering

ABSTRACT

With the widespread applications of lithium-ion batteries in high-end emerging industries, monitoring the state of charge (SOC) of lithium-ion batteries has attracted extensive attention and research. However, accurate SOC estimation is challenging due to the variability of lithium-ion batteries under different operating conditions. In this paper, a hybrid model is proposed for precise SOC estimation of lithium-ion batteries. The hybrid model combines a memory-enhanced gated recurrent unit (ME-GRU) network with an adaptive unscented Kalman filter (AUKF), and optimizes the hyperparameters of the ME-GRU network using a covariance matrix adaptation evolution strategy (CMAES). Specifically, the ME-GRU network serves as a deep learning-based method to produce preliminary SOC estimation results, avoiding the complex battery modeling process required by AUKF. In addition, AUKF corrects the preliminary SOC estimation results to mitigate the negative effect of inherent noise in the data. Moreover, during the optimization process, we introduce Gaussian process into CMAES, which effectively reduces the time required and enhances the real-time performance of the hybrid model in SOC estimation. This optimization process avoids the time-consuming hyperparameter design of the ME-GRU network, and helps the hybrid model yield accurate SOC estimation under different operating conditions. Experiments on an individual battery dataset and a battery pack dataset demonstrate that the hybrid model achieves satisfactory SOC estimation, with the root mean square error (RMSE) below 0.6% and the maximum error below 2.5% across both datasets.

1. Introduction

In the field of energy, lithium-ion batteries have the advantages of high power density, extended cycle life, and cost-effectiveness, and have been widely used in cutting-edge industries such as new energy vehicles, smart grids, and rail transportation [1–3]. During the operation of lithium-ion batteries, the state of charge (SOC) serves as a key metric, which reflects the available battery capacity [4]. Inaccurate SOC estimation poses significant risks, potentially leading to battery overcharge, overdischarge, and even catastrophic outcomes such as fire and explosion [5]. Hence, precise SOC estimation is crucial to ensure battery safety [6,7].

A large number of SOC estimation methods have been proposed. Generally, they can be divided into four categories: characteristic parameters (CP)-based methods [8–10], Coulomb counting (CC)-based method [11,12], model-based methods [13], and deep learning-based methods [14,15]. Next, we briefly introduce them.

CP-based methods, such as the electrochemical impedance spectroscopy (EIS) method and the open circuit voltage (OCV) method, directly estimate SOC through parameter characterization. However, the EIS method relies on precise impedance measurements, which is very difficult [8]. The OCV method depends on OCV-SOC lookup tables, which may vary significantly under different temperatures and operating conditions [9,10].

CC-based method uses a current sensor to measure the discharge current and updates the battery SOC by subtracting the elapsed charge [11]. CC-based method is easy to implement, and its SOC calculation results are less affected by noise. However, it operates in an open-loop estimation manner and is susceptible to cumulative errors when the load current measurement is inaccurate. In addition, inaccurate initial SOC values can further aggravate the error effect, which greatly limits its effectiveness in scenarios where high-precision SOC is required [12].

Model-based methods usually characterize the dynamic electrochemical process of battery charging and discharging by creating a battery model, and then calculate the battery SOC based on the established model. In these methods, the battery model is commonly formulated as a state observer, and the closed-loop feedback control is used to overcome the cumulative error in CC-based method [16].

* Corresponding author.
E-mail addresses: wangwei729@csu.edu.cn (W. Wang), ywang@csu.edu.cn (Y. Wang), zijzhang@cityu.edu.hk (Z. Zhang).

https://doi.org/10.1016/j.est.2025.115948
Received 25 July 2024; Received in revised form 1 February 2025; Accepted 21 February 2025
Available online 27 March 2025

In existing model-based methods, the Kalman filter (KF) is widely utilized for state observation, which is known for its capability to correct initial errors and mitigate system noise. For instance, Zhao et al. [17] proposed the extended KF (EKF) to estimate battery SOC, which performs first-order linear truncation on the Taylor expansion of the nonlinear battery system. Nevertheless, in the linearization process, EKF only retains first-order terms in the Taylor expansion and discards second-order and higher-order terms, which may introduce linearization errors and lead to poor filtering performance. Tian et al. [18] and Xiong et al. [19] employed the unscented KF (UKF) to achieve precise SOC estimation. UKF approximates the posterior probability density of the state by employing a set of deterministic samples, which can yield better filtering performance than EKF. However, UKF fixes two key parameters, i.e., the state noise covariance and the measurement noise covariance, which may result in significant fluctuations in the estimation results.

In addition to the above KF-related methods, some researchers have also investigated other filtering methods for SOC estimation. Alćantara et al. [20] developed a Gaussian sum filtering method for estimating SOC of batteries, which offers higher estimation accuracy than EKF and provides information about the probability density function of SOC. Wang et al. [21] proposed a framework for observing the battery's SOC using an unscented particle filter. This framework incorporates a recursive prediction method that takes historical data into account, with the aim of improving SOC estimation accuracy. Chen et al. [22] presented an enhanced H_∞ filtering method based on the reverse recursion of historical data, ensuring that SOC estimation errors of aging batteries remain within 3%. However, the above model-based methods heavily rely on the precision of battery modeling, and complex electrochemical reactions make battery modeling very challenging. Moreover, the model parameters are significantly influenced by complex operating conditions and battery aging, which may also affect the precision of battery modeling [23,24].

In contrast, deep learning-based methods do not need an in-depth understanding of battery structure or modeling. Instead, they implicitly learn the nonlinear relationships (including electrochemical reactions, aging processes, etc.) between the system states and the test variables through feature extraction [25,26]. To achieve accurate battery SOC estimation, deep learning-based methods need sufficient training and testing data, along with appropriate hyperparameter definitions [27].

Among deep learning-based methods, the most popular methods include recurrent neural network (RNN) and its variants, such as long short-term memory (LSTM) network and gated recurrent unit (GRU) network. For instance, Chaoui et al. [28] employed RNN for online analysis of electric vehicle batteries. However, RNN tends to lose historical information during the long-term dependency modeling process and may encounter gradient vanishing or explosion. LSTM introduces a gating mechanism and memory cells, which addresses the shortcomings of RNN to a certain degree [29]. Eleftheriadis et al. [30] and Zhang et al. [31] employed stacked LSTMs for SOC estimation, effectively capturing the nonlinear dynamic relationship between battery input and output SOC. Nevertheless, the high computational complexity and hyperparameter design of LSTM result in long training time, hindering its ability to meet real-time requirements. In contrast, GRU employs update and reset gates to control the flow of historical information, reducing parameters through fewer gate structures. This enhances computational efficiency without compromising accuracy. Recently, Hannan et al. [32] applied GRU to battery SOC estimation, yielding promising results. However, GRU merges the cell state with the hidden state, which may lead to the loss of long-range information in some cases. Moreover, the above deep learning-based methods share two common limitations in SOC estimation. Firstly, due to the data-driven nature of deep learning-based methods, the inherent noise within the data inevitably causes estimation errors [33]. Secondly, their performance is significantly impacted by the quality of hyperparameters. Under this condition, manual design of hyperparameters for different battery conditions is required, which is time-consuming.

Based on the above analysis, model-based methods excel at noise suppression but require a complex battery modeling process. In addition, deep learning-based methods possess strong feature learning capabilities but are easily affected by noise in the data. To leverage the strengths of these two kinds of methods and overcome their respective limitations, this paper develops a hybrid model that combines a memory-enhanced GRU (ME-GRU) network with an adaptive UKF (AUKF). On the one hand, the ME-GRU network serves as a deep learning-based method to produce preliminary SOC estimation results, avoiding the complex battery modeling process required by AUKF. Moreover, given the time-series nature of battery SOC estimation, we design a memory enhancement unit in the ME-GRU network to effectively improve the precision of the preliminary SOC estimation results. On the other hand, AUKF corrects the preliminary SOC estimation results to mitigate the negative effect of inherent noise in the data. Moreover, to further address the time-consuming hyperparameter design problem, a Gaussian process (GP)-based covariance matrix adaptation evolution strategy (CMAES) [34], called GP-CMAES, is designed to adaptively optimize the hyperparameters of the ME-GRU network. The main contributions of this paper are outlined as follows:

(1) In the ME-GRU network, a memory enhancement unit based on window sliding is designed to reconstruct the input for each time step and integrate information from historical time steps. With the memory enhancement unit, the ME-GRU network strengthens its ability to capture long-term dependencies in time-series data, thereby improving the accuracy of preliminary SOC estimation. As a result, the ME-GRU network is well-suited for battery SOC estimation based on time-series data.

(2) In GP-CMAES, GP is adopted to reduce the time required and enhance the real-time performance of the model in SOC estimation. The introduction of GP-CMAES not only automates the hyperparameter tuning process, reducing time costs, but also enhances the robustness and generalization ability of the ME-GRU network. Overall, GP-CMAES can further improve the practicality of the ME-GRU network under various battery operating conditions.

(3) AUKF monitors the prediction error to enable real-time correction of both the state and measurement noise covariances, eliminating the need for manual design of the noise covariance matrices. Additionally, this approach demonstrates strong noise suppression capabilities. In practical applications, AUKF further enhances the adaptability and accuracy of the ME-GRU network for SOC estimation across various operating conditions.

The subsequent structure is as follows. Section 2 describes the details of the proposed hybrid model. Section 3 analyzes the performance of the proposed hybrid model. Finally, Section 4 summarizes the full paper.

2. Methodology

The framework of the proposed hybrid model is depicted in Fig. 1. It comprises three main components: the ME-GRU network, GP-CMAES, and AUKF. The overall process of the proposed hybrid model is as follows. Firstly, the ME-GRU network is constructed, in which the initial hyperparameters are optimized by using GP-CMAES. Secondly, the current, voltage, and temperature data are fed into the ME-GRU network to obtain the preliminary SOC estimation results (denoted as SOC_{GRU}). Finally, based on the SOC calculation results of CC-based method (denoted as SOC_{CC}), SOC_{GRU} are further corrected by AUKF to obtain more accurate SOC estimation results (denoted as SOC_{est}).

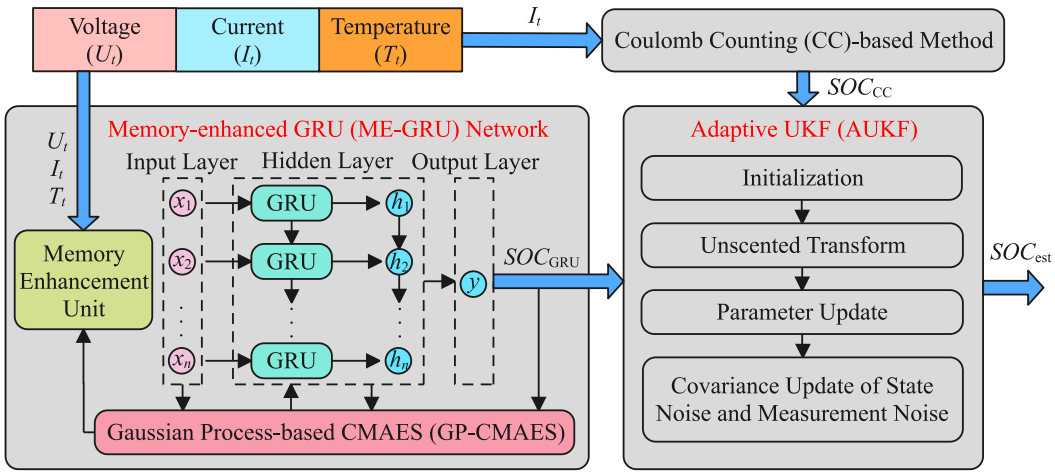


Fig. 1. The overall framework of the proposed hybrid model.

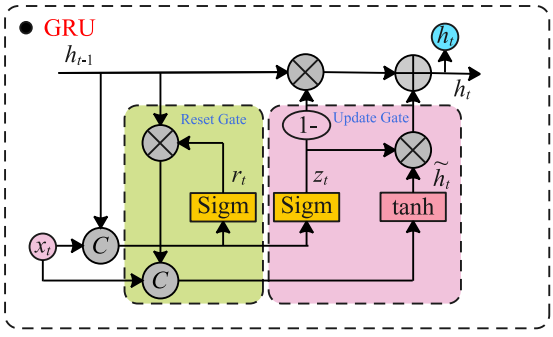


Fig. 2. The structure of the GRU unit.

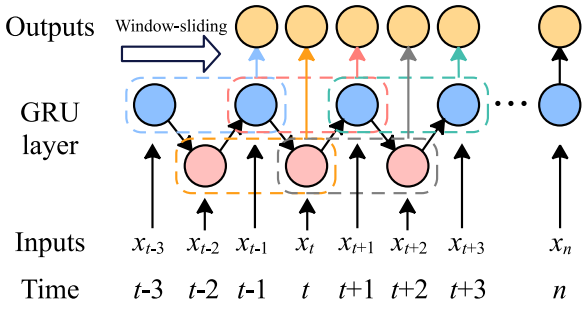


Fig. 3. The structure of the memory enhancement unit.

2.1. Structure of the ME-GRU network

As a variant of RNN, GRU can handle the long-term dependencies within time series data, making it suitable for battery SOC estimation. The structure of the GRU unit is given in Fig. 2. From Fig. 2, the GRU unit adopts two gates, i.e., the reset gate and the update gate, to control the flow of historical information. The mathematical expressions of the GRU unit are as follows:

$$r_t = sigmoid(W_r \cdot [h_{t-1}, x_t]) \tag{1}$$

$$\tilde{h}_t = tanh(W \cdot [r_t \odot h_{t-1}, x_t]) \tag{2}$$

$$z_t = sigmoid(W_Z \cdot [h_{t-1}, x_t]) \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{4}$$

Next, we introduce the reset gate and the update gate.

Reset gate: It determines the weights of input information at the current time step t and information from the previous time step $(t-1)$. Specifically, it begins by concatenating the input x_t with the hidden state h_{t-1}. The concatenated result is then multiplied by the weight matrix W_r and passed through the sigmoid function, yielding the reset gate control signal r_t, as shown in Eq. (1). Subsequently, r_t is used to perform element-wise multiplication (Hadamard product) with h_{t-1} and then concatenated with x_t. The concatenated result is passed through the $tanh$ activation function to produce the current candidate hidden state $\tilde{h}_t$, which ranges from [−1, 1], as shown in Eq. (2). Note that when r_t approximates 0, it indicates that most of h_{t-1} is discarded. Conversely, if r_t is close to 1, it suggests a tendency to retain the majority of h_{t-1}. It can be seen that the reset gate plays a crucial role in capturing short-term dependencies within time series data.

Update gate: It determines how much of historical information should be transmitted to the current time step. The control signal z_t, obtained from Eq. (3), determines how the current hidden state h_t balances the preservation relationship between h_{t-1} and $\tilde{h}_t$. Eq. (4) outlines the computation of h_t. If the update gate control signals $[z_{t_i}, \ldots, z_{t-1}]$ from multiple previous time steps are all approximately 0 or 1, then the input from time steps t_i to $(t-1)$ is almost or scarcely transmitted to h_t. In fact, the update gate achieves the continuous preservation and transmission of h_{t_i} from t_i to t. The design of the update gate effectively mitigates the issues of gradient vanishing and exploding in RNNs, and is beneficial for capturing long-term dependencies in time series data.

In order to make full use of the information from historical time steps, we design a memory enhancement unit based on window sliding, as shown in Fig. 3. This unit implements an input reconstruction process with a sliding window size s. In this paper, we select one time step from every two adjacent time steps, with the aim of expanding the coverage of the sliding window and mitigating output fluctuations. Therefore, the reconstructed input at t can be represented as $[x_t, x_{t-2}, \ldots, x_{t-s}]$, where s is a multiple of 2. Precisely, the memory enhancement unit ensures that the reconstructed input at the current time step includes the information from historical time steps, thereby effectively utilizing historical data. This significantly enhances the network's memory capability. In this paper, x_t is defined as $x_t = [V_t, I_t, T_t]$, where V_t, I_t, and T_t represent the battery voltage, current, and temperature data, respectively.

In the training of the ME-GRU network, the RMSE is employed as the loss function, which quantifies the difference between the estimated SOC and the actual SOC during the forward propagation process. The adaptive moment estimation optimizer is utilized to iteratively update

Algorithm 1 Implementation process of GP-CMAES

Input: l, g_{max}, and $MaxNE$
Output: the best solution $\mathbf{a}^* = (a_1^*, a_2^*, a_3^*, a_4^*, a_5^*)$ in $\mathbb{DB}$
Initialize $\mathbf{m}(g), \mathbf{C}(g), \mathbf{p}(g), \mathbf{o}(g)$, and $\xi(g)$;
Generate l solutions by Latin Hypercube Sampling and evaluate them by Eq. (5);
$NE = l$;
Initialize database (denoted as $\mathbb{DB}$) with these l solutions;
while $NE < MaxNE$ **do**
  Build GP based on $\mathbb{DB}$;
  for $g = 1 : g_{max}$ **do**
    for $i = 1 : \epsilon$ **do**
      $\mathbf{s}_i(g) = \mathbf{m}(g) + \xi(g)(\mathbf{C}(g))^{\frac{1}{2}}\mathcal{N}(\mathbf{0}, \mathbf{I})$;
    end for
    Evaluate and sort $\mathbf{s}_1(g), \ldots, \mathbf{s}_\epsilon(g)$ in ascending order by GP;
    Select the first k solutions to calculate $\mathbf{m}(g+1), \mathbf{C}(g+1), \mathbf{p}(g+1), \mathbf{o}(g+1)$, and $\xi(g+1)$ via Eq. (13) and Eqs. (15)–(18);
  end for
  Evaluate the best solution by Eq. (5) and put it into $\mathbb{DB}$;
  $NE = NE + 1$;
end while
Output the best solution in $\mathbb{DB}$.

the weight matrices and bias vectors of the network based on the gradient of the loss function. Although a deep ME-GRU network may potentially yield better performance, it will increase computational time and may lead to overfitting. To improve computational efficiency, we set a hyperparameter (i.e., the number of the hidden layers) of the ME-GRU network to 1. For the remaining hyperparameters, we model the setting of them as an optimization problem as follows:

$$\begin{aligned} min: & f(\mathbf{a}) = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(SOC_{\mathrm{act},t}(\mathbf{a}) - SOC_{\mathrm{GRU},t}(\mathbf{a})\right)^2} \\ \text{s.t.} \quad & \mathbf{a} = (a_1, a_2, a_3, a_4, a_5) \\ & a_1 \in [2, 20] \\ & a_2 \in [50, 150] \\ & a_3 \in [50, 200] \\ & a_4 \in [0.0001, 0.01] \\ & a_5 \in [0.05, 0.15] \end{aligned} \tag{5}$$

where n is the number of time steps, $a_1 - a_5$ represent the remaining hyperparameters (i.e., the sliding window size, the number of neurons, the max epochs, the initial learning rate, and the learning rate drop factor), and SOC_{act} denotes the actual SOC values.

This optimization problem aims to minimize $f(\cdot)$ (i.e., RMSE) by adjusting the hyperparameters. To solve it, this paper proposes GP-CMAES.

2.2. GP-CMAES

CMAES is a famous numerical optimization strategy, which is widely used to solve nonlinear, non-convex, and black-box optimization problems. In the lithium-ion battery SOC estimation problem, CMAES can be used to adaptively optimize the hyperparameters of the ME-GRU network, which can not only reduce the estimation error, but also be applicable to various operating conditions. Standard CMAES generates offspring solutions by sampling from a Gaussian distribution and updates the Gaussian distribution in each iteration. However, this process requires multiple evaluations, and each evaluation is time-consuming. It may not meet the real-time requirements of SOC estimation. To address this issue, GP-CMAES is proposed, which effectively reduces the number of actual RMSE evaluations (denoted as NE) in the optimization process and consequently decreases the overall optimization time. The process of GP-CMAES is described in **Algorithm 1**, and the main steps are described as follows:

(1) Initialization: At first, GP-CMAES initializes five elements: mean vector $\mathbf{m}(g)$, covariance matrix $\mathbf{C}(g)$, evolution paths $\mathbf{p}(g)$ and $\mathbf{o}(g)$, and mutation strength $\xi(g)$. Subsequently, l solutions are generated by Latin Hypercube Sampling, and are subsequently input into the ME-GRU network for actual RMSE evaluations. Then, NE is set to l, and a database $\mathbb{DB}$ is created with these l solutions.

(2) GP establishment: Based on $\mathbb{DB} = \{(s_i, f(s_i)) | i = 1, \ldots, l\}$, we assume that the objective function has the form of:

$$f(s) = \mu + L(s) \tag{6}$$

where μ is the mean of GP, $L(s)$ is a GP with zero mean, variance σ^2, and the correlation function $Z(s_i, s_j; \theta)$ $(i, j \in [1, l])$. This paper uses Gaussian correlation function:

$$Z\left(s_i, s_j; \theta\right) = \exp\left[-\sum_{i=1}^{d}\theta_i\left(s_i - s_j\right)^2\right] \tag{7}$$

where d is the dimension of the optimization problem, and $\theta = (\theta_1, \ldots, \theta_d)$ is hyperparameter set. The hyperparameters in θ can be estimated by maximizing the likelihood:

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^d}{\operatorname{argmax}}\left(-\frac{n}{2}\ln\hat{\sigma}^2 - \frac{1}{2}\ln|Z|\right) \tag{8}$$

where

$$\hat{\mu} = \frac{I^T B^{-1} f}{I^T B^{-1} I} \tag{9}$$

$$\hat{\sigma}^2 = \frac{(f - 1\hat{\mu})^T B^{-1}(f - 1\hat{\mu})}{n} \tag{10}$$

In Eqs. (9) and (10), B is an $l \times l$ correlation matrix with entry $Z_{ij} = Z(s_i, s_j; \theta)$, $f = (f(s_1), \ldots, f(s_l))^T$, and I is an $l \times 1$ vector of ones. After the hyperparameters are learned, the best linear unbiased prediction of GP can be determined with the estimated hyperparameters:

$$\hat{f}(s) = \hat{\mu} + \tau^T B^{-1}(f - I\hat{\mu}) \tag{11}$$

where τ is an $l \times 1$ correlation vector with entry $\tau_i = Z(s, s_i; \hat{\theta})$.

(3) Optimization: CMAES is run to solve Eq. (5) for g_{max} generations. During this process, since GP learns the inherent property of the objective function of Eq. (5), we use GP as an alternative to evaluate quality of the offspring solutions, with the aim of saving the number of actual RMSE evaluations.

To be specific, in the gth iteration ($g \in [1, g_{max}]$), ϵ offspring solutions are generated through the Gaussian distribution:

$$\mathbf{s}_i(g) = \mathbf{m}(g) + \xi(g)(\mathbf{C}(g))^{\frac{1}{2}}\mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{12}$$

Afterward, these ϵ offspring solutions are sorted in ascending order through the GP evaluation. The first k solutions are selected and used to update $\mathbf{m}(g)$ for the next generation:

$$\mathbf{m}(g+1) = \mathbf{m}(g) + \sum_{i=1}^{k} F_i\left(\mathbf{s}_i(g) - \mathbf{m}(g)\right) \tag{13}$$

where F_i is the weight corresponding to the ith solution. F_i can be calculated as follows:

$$F_i = \frac{\log(k + 0.5) + \log\left(\mathrm{rank}_i\right)}{k\log(k + 0.5) + \sum_{i=1}^{k}\log\left(\mathrm{rank}_i\right)} \tag{14}$$

where rank_i is the ranking of the ith solution.

Next, the evolution path for $\mathbf{C}(g)$ (denoted as $\mathbf{p}(g)$) is updated via a long-term average of consecutive steps of the mean vector:

$$\mathbf{p}(g+1) = \left(1 - c_p\right)\mathbf{p}(g) + \sqrt{c_p\left(2 - c_p\right)}k_{\mathrm{eff}}\frac{(\mathbf{m}(g+1) - \mathbf{m}(g))}{\xi(g)} \tag{15}$$

where $c_p \in (0, 1)$ is the learning rate and $k_{\mathrm{eff}} = 1/\sum_{i=1}^{k} w_i^2$ is the effective sample size.

Subsequently, $\mathbf{C}(g)$ is updated as follows:

$$\mathbf{C}(g+1) = \left(1-c_1\right)\mathbf{C}(g) + c_1\mathbf{p}(g+1)(\mathbf{p}(g+1))^{\mathrm{T}} \tag{16}$$

where $c_1 \in (0,1)$ is the learning rate.

In addition, the evolution path for $\boldsymbol{\xi}(g)$ (denoted as $\mathbf{o}(g)$) is updated as follows:

$$\mathbf{o}(g+1) = \left(1-c_o\right)\mathbf{o}(g) + \sqrt{c_o\left(2-c_o\right)}\sqrt{k_{\mathrm{eff}}}\mathbf{C}(g)^{-\frac{1}{2}}\frac{(\mathbf{m}(g+1)-\mathbf{m}(g))}{\boldsymbol{\xi}(g)} \tag{17}$$

where $c_o > 0$ is the learning rate. Then, ξ_g is updated as follows:

$$\boldsymbol{\xi}(g+1) = \boldsymbol{\xi}(g)\cdot e^{\frac{1}{d_\xi}\left(\frac{\|\mathbf{o}(g+1)\|}{\mathbb{E}[\|\mathcal{N}(\mathbf{0},\mathbf{I})\|]}-1\right)} \tag{18}$$

where $d_\xi \geq 0$ is the damping parameter.

After g_{max} generations, we select the best solution predicted by GP. Then, this solution is input into the ME-GRU network for one actual RMSE evaluation and is stored into $\mathbb{DB}$. Afterward, NE is added by 1.

Steps (2) and (3) are repeated until the available number of actual RMSE evaluations (denoted as $MaxNE$) is reached. Finally, the best solution in $\mathbb{DB}$ is output.

In this paper, the value ranges of l, g_{max}, and $MaxNE$ were determined based on [35–38] and then tuned using the *irace* package [39]. Finally, l, g_{max}, and $MaxNE$ were set to 24, 30, and 150, respectively.

2.3. AUKF

KF can obtain the optimal estimate of state based on the state space model of the Gaussian white noise system. Various improvements of KF, including EKF and UKF, have proven the effectiveness of addressing noise-related challenges within nonlinear systems. Compared with EKF, UKF employs a Gaussian approximation approach, enhancing its capability to handle nonlinear functions. Therefore, UKF may be more suitable for battery SOC estimation problem.

In standard UKF, the noise covariance is typically treated as a constant value. However, the statistical characteristics of state noise and measurement noise remain elusive during battery operation. Relying on fixed noise parameters for system state estimation may lead to considerable errors. Therefore, in order to meet the adaptability and accuracy of the battery SOC estimation under different operating conditions, AUKF [40] is adopted.

The core concept of AUKF is to use Unscented Transform (UT) to handle the nonlinear propagation of the mean and covariance. Specifically, UT selects some sample points (called sigma points) from the original state distribution according to a certain rule, ensuring that the mean and variance of these sigma points match those of the original state distribution. These sigma points are then passed through the nonlinear function to obtain the transformed means and covariances. Against the standard UKF, AUKF introduces an adaptive noise covariance method. This method realizes real-time correction of the model's noise covariance by monitoring prediction errors. Consequently, AUKF can dynamically adjust the filtering parameters based on the noise uncertainty. We assume that the state space equation of the nonlinear system is:

$$\begin{cases} x_t = \phi(x_{t-1}) + \omega_t \\ y_t = \psi(x_t) + \upsilon_t \end{cases} \tag{19}$$

where $\phi(\cdot)$ and $\psi(\cdot)$ are the nonlinear functions, x_t and y_t are the system state vector and the measurement vector at time t, respectively, and ω_t and υ_t are state noise and measurement noise of the system at time t, respectively.

The specific recursive processes of AUKF are shown as follows:

(1) Initialization:

$$\begin{cases} \hat{x}_0^* = E(x_0) \\ P_0^* = E\left[\left(x_0-\hat{x}_0^*\right)\left(x_0-\hat{x}_0^*\right)^T\right] \end{cases} \tag{20}$$

where $\hat{x}_0^*$ and P_0^* are the initial state estimation and the error covariance, respectively, $[\cdot]^*$ represents the posteriori values, and E(·) represents the mean value.

(2) Calculating the sigma points and the weight coefficients:

$$\chi_{i,t-1} = \begin{cases} \hat{x}_{t-1}^*, i = 0 \\ \hat{x}_{t-1}^* + \left(\sqrt{(d+\lambda)P_{t-1}^*}\right)_i, i = 1,\ldots,d \\ \hat{x}_{t-1}^* - \left(\sqrt{(d+\lambda)P_{t-1}^*}\right)_i, i = d+1,\ldots,2d \end{cases} \tag{21}$$

$$\begin{cases} \omega_i^m = \frac{\lambda}{\lambda+d}, i = 0 \\ \omega_i^c = \frac{\lambda}{\lambda+d} + \left(1-\alpha^2+\beta\right), i = 0 \\ \omega_i^m = \omega_i^c = \frac{\lambda}{2(\lambda+d)}, i = 1,\ldots,2d \end{cases} \tag{22}$$

where $\hat{x}_{t-1}^*$ and P_{t-1}^* represent the state estimation and the error covariance at time step $(t-1)$, respectively, d is the dimension of the state vector, $[\cdot]_i$ represents the ith column of the matrix, ω_i^m and ω_i^c are the weight factors of the state estimation and the error covariance at sigma point i, respectively, $\lambda = 3\alpha^2 - d$ is the scale factor, $\alpha \in (1e{-}4, 1)$ defines the spread of the sigma points around the state vector, and β is the nonnegative weight coefficient.

(3) Updating the prior estimation:

Project the sigma points according to the state function:

$$\chi_{i,t|t-1} = \phi(\chi_{i,t-1}) \tag{23}$$

Calculate the prior estimation of the system state and the prior error covariance:

$$\hat{x}_t = \sum_{i=0}^{2d}\omega_i^m\chi_{i,t|t-1} \tag{24}$$

$$P_t = \sum_{i=0}^{2d}\omega_i^c\left[\chi_{i,t|t-1}-\hat{x}_t\right]\left[\chi_{i,t|t-1}-\hat{x}_t\right]^T + Q_{t-1} \tag{25}$$

where $t|t-1$ represents the prediction values at time step t in condition of $(t-1)$, and Q_{t-1} represents state noise covariance matrix at time step $(t-1)$.

(4) Updating measurement estimation:

Project the sigma points according to the measurement function:

$$\gamma_{i,t|t-1} = \psi\left(\chi_{i,t|t-1}\right) \tag{26}$$

Calculate the predicted value of the measurement estimation and the covariance:

$$\hat{y}_t = \sum_{i=0}^{2d}\omega_i^m\gamma_{i,t|t-1} \tag{27}$$

$$P_{y,t} = \sum_{i=0}^{2d}\omega_i^c\left[\gamma_{i,t|t-1}-\hat{y}_t\right]\left[\gamma_{i,t|t-1}-\hat{y}_t\right]^T + R_{t-1} \tag{28}$$

where R_{t-1} represents measurement noise covariance matrix at time step $(t-1)$.

Calculate the cross-covariance of the state estimation and measurement estimation:

$$P_{xy,t} = \sum_{i=0}^{2d}\omega_i^c\left[\chi_{i,t|t-1}-\hat{x}_t\right]\left[\gamma_{i,t|t-1}-\hat{y}_t\right]^T \tag{29}$$

(5) Updating the Kalman gain, posteriori state estimation, and error covariance:

$$K_{kal,t} = P_{xy,t}P_{y,t}^{-1} \tag{30}$$

$$\hat{x}_t^* = \hat{x}_t + K_{kal,t}\left(y_t-\hat{y}_t\right) \tag{31}$$

$$P_t^* = P_t - K_{kal,t}P_{y,t}K_{kal,t}^T \tag{32}$$

(6) Updating noise covariance adaptively:

$$Q_t = K_{kal,t}H_tK_{kal,t}^T \tag{33}$$

$$R_t = \sum_{i=0}^{2d} \omega_i^c \left[\gamma_{i,t|t-1} - y_t\right]\left[\gamma_{i,t|t-1} - y_t\right]^T + H_t \tag{34}$$

where

$$H_t = \sum_{i=t-L_w+1}^{t} b_i b_i^T \tag{35}$$

where b_i is the residuals of the estimated states at time step i, and $L_w = 60$ is the window size for covariance matching.

In this study, the primary purpose of employing AUKF is to effectively filter the noise from SOC_{GRU}, consequently enhancing the accuracy of SOC estimation. Initially, we establish a SOC state space model. Then, we use SOC_{CC} as the system state variable, SOC_{GRU} as the measurement variable, and the battery current as the input variable. Subsequently, these three variables are integrated into the AUKF recursion formula, yielding a refined SOC estimate through the filtering process. Note that, the SOC state space model is formulated as follows:

$$\begin{cases} SOC_{\mathrm{CC},t} = SOC_{\mathrm{CC},t-1} - \eta \frac{I_t \Delta t}{C_n} + w_t \\ y_t = SOC_{\mathrm{GRU},t} + v_t \end{cases} \tag{36}$$

where η indicates Coulomb efficiency, I_t represents current, Δt denotes the sampling interval, C_n means the nominal capacity, and $w_t \sim N(0, Q)$ and $v_t \sim N(0, R)$ denote the state and measurement noise, respectively.

3. Experiment results and analysis

This section details the public dataset of one lithium-ion battery cell and the collected actual operation dataset of two lithium-ion battery packs. On these two datasets, we verified the superiority of the proposed hybrid model in SOC estimation by comparing with the GRU network and the GRU+UKF model. It is worth noting that the GRU network refers to the ME-GRU network optimized using GP-CMAES.

3.1. Datasets

The first dataset is the public A123 battery cell dataset provided by the University of Maryland Data Storage Repository. To better simulate real driving conditions of electric vehicles, we considered three dynamic driving modes: Dynamic Stress Test (DST), US06 driving cycle, and Federal Urban Driving Schedule (FUDS). DST simulates complex road driving conditions, US06 driving cycle simulates highway driving conditions, and FUDS simulates urban driving conditions. The data from DST forms the training set, while the data from US06 and FUDS constitute the test sets. Additionally, the driving conditions of DST, US06, and FUDS were conducted at different temperatures of 0 °C, 20 °C, 30 °C, and 50 °C.

The second dataset is from an electric vehicle company. We collected the current, voltage, and temperature data from two distinct types of electric vehicle battery packs located in Shanghai and Hefei. These data span the time frame from December 2022 to April 2023 and constitute the battery pack dataset.

The specifications of the above two datasets are summarized in Table 1.

To accelerate the convergence speed of the proposed hybrid model, alleviate the influence of singular sample data, and mitigate the risk of overfitting, a normalization process was applied to the data. This process maps the data into [−1, 1], as illustrated in Eq. (37):

$$x_{norm} = \frac{2(x - x_{avg})}{x_{max} - x_{min}} \tag{37}$$

where x_{max} and x_{min} denote the maximum and minimum values of the data, respectively, x_{avg} denotes the average value, and x denotes the battery data.

Table 1
Parameters of one battery cell and two battery packs. Note that, Pack 1 has two branches, each consisting of 13 battery cells connected in series. Pack 2 has one branch, consisting of 8 battery cells connected in series.

Type	Parameter	Value
Cell	Cell Chemistry	LiFePO4
	Capacity Rating	1.1 Ah
	Maximum Voltage	3.6 V
	Discharge Cut-off Voltage	2 V
Pack 1	Cell Chemistry	LiFePO4
	Number of Series-Parallel cells	26 Series and 2 Parallel
	Cell Voltage	3.2 V
	Nominal Voltage	83.2 V
	Nominal Current	400 A
	Nominal Capacity	405 Ah
Pack 2	Cell Chemistry	LiFePO4
	Number of Series-Parallel cells	8 Series and 1 Parallel
	Cell Voltage	3.2 V
	Nominal Voltage	25.6 V
	Nominal Current	206 A
	Nominal Capacity	206 Ah

Table 2
SOC estimation results for US06 at 0 °C, 20 °C, 30 °C, and 50 °C.

US06	RMSE(%)			MAX(%)		
	GRU	GRU+ UKF	Proposed model	GRU	GRU+ UKF	Proposed model
0 °C	2.05	1.18	**0.52**	10.83	5.27	**2.40**
20 °C	2.02	1.03	**0.47**	13.54	3.56	**1.97**
30 °C	2.18	1.08	**0.50**	13.31	3.23	**2.09**
50 °C	2.14	1.05	**0.51**	13.96	3.79	**2.04**

Table 3
SOC estimation results for FUDS at 0 °C, 20 °C, 30 °C, and 50 °C.

FUDS	RMSE(%)			MAX(%)		
	GRU	GRU+ UKF	Proposed model	GRU	GRU+ UKF	Proposed model
0 °C	1.44	0.86	**0.41**	8.12	3.31	**2.02**
20 °C	1.93	0.89	**0.39**	9.73	3.60	**1.63**
30 °C	2.46	1.10	**0.42**	9.94	3.71	**2.06**
50 °C	2.01	1.09	**0.39**	7.71	3.52	**1.99**

In this study, RMSE and MAX were selected as evaluation metrics. The equations of them are given as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(SOC_{\mathrm{act},t} - SOC_{\mathrm{est},t}\right)^2} \tag{38}$$

$$\mathrm{MAX} = \max\left(\left|SOC_{\mathrm{act},t} - SOC_{\mathrm{est},t}\right|\right) \tag{39}$$

3.2. SOC estimation with the proposed hybrid model in A123 battery cell

In this subsection, we utilized the DST data as the training set and the US06 and FUDS data as the test sets to evaluate the SOC estimation performance of the hybrid model proposed in this paper. Subsequently, the evaluation results were compared with those of the GRU network and the GRU+UKF model. Figs. 4 and 5 respectively present the results of the three models at different temperatures for US06 and FUDS, while Tables 2 and 3 provide a summarized comparative analysis. Note that, the initial SOC during discharge is assumed to be 100% in this analysis.

The results indicate that the GRU network can capture the downward trend of SOC, with the RMSE values within 3% on the US06 and DST test sets. However, this error is not satisfactory for the SOC estimation problem. Furthermore, the MAX values of the GRU network are approximately 10% on both test sets, indicating significant fluctuations in the estimation and making it challenging to meet practical application requirements. To address this issue, the estimation results

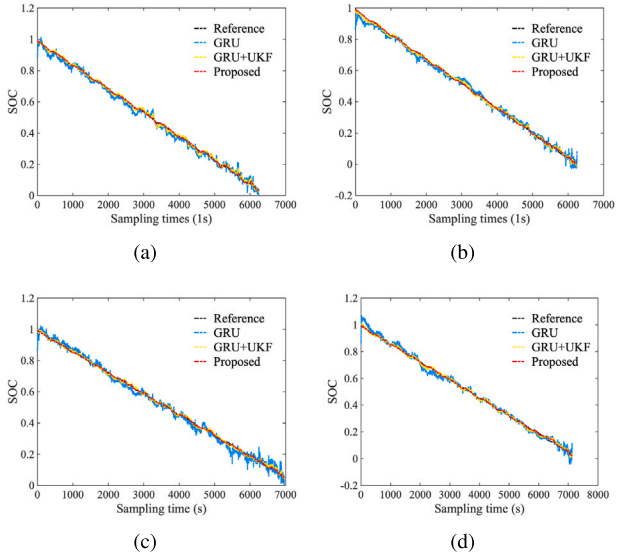


Fig. 4. Results for US06 at: (a) 0 °C; (b) 20 °C; (c) 30 °C; (d) 50 °C.

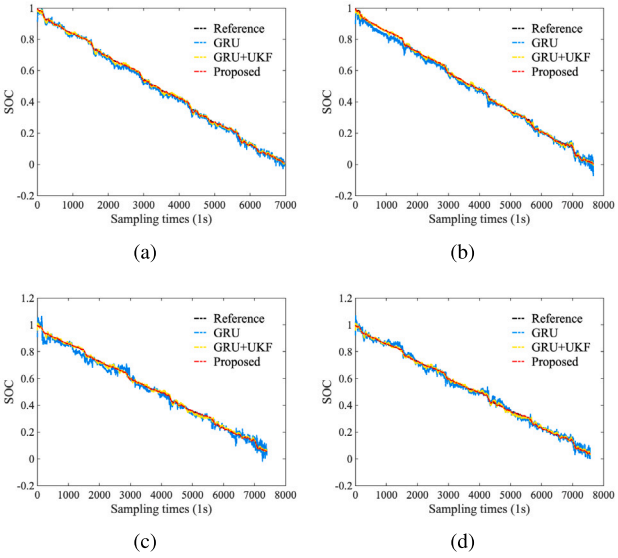


Fig. 5. Results for FUDS at: (a) 0 °C; (b) 20 °C; (c) 30 °C; (d) 50 °C.

of the GRU network can be corrected by using UKF, which can reduce the RMSE and MAX values to some extent. The proposed hybrid model in this paper adopts AUKF, which combines UKF with an adaptive state noise update, further reducing estimation fluctuations and improving estimation accuracy. Specifically, the proposed hybrid model achieves the best results in terms of RMSE and MAX at all the considered temperatures. For example, under the US06 cycle at 20 °C, the RMSE value of the proposed hybrid model (0.47%) significantly outperforms that of the GRU network (2.02%) and the GRU+UKF model (1.03%). Under the FUDS cycle at 20 °C, the proposed hybrid model achieves the best RMSE value of 0.39%, surpassing the GRU network (1.93%) and the GRU+UKF model (0.89%). Additionally, under both the US06 and FUDS cycles at 20 °C, the MAX values decrease from 13.54% to 1.97% and from 9.73% to 1.63%, respectively. Overall, the proposed hybrid model maintains the RMSE values of SOC estimation within 0.6% and the MAX values within 2.5%.

Fig. 6. An EALV and its battery pack.

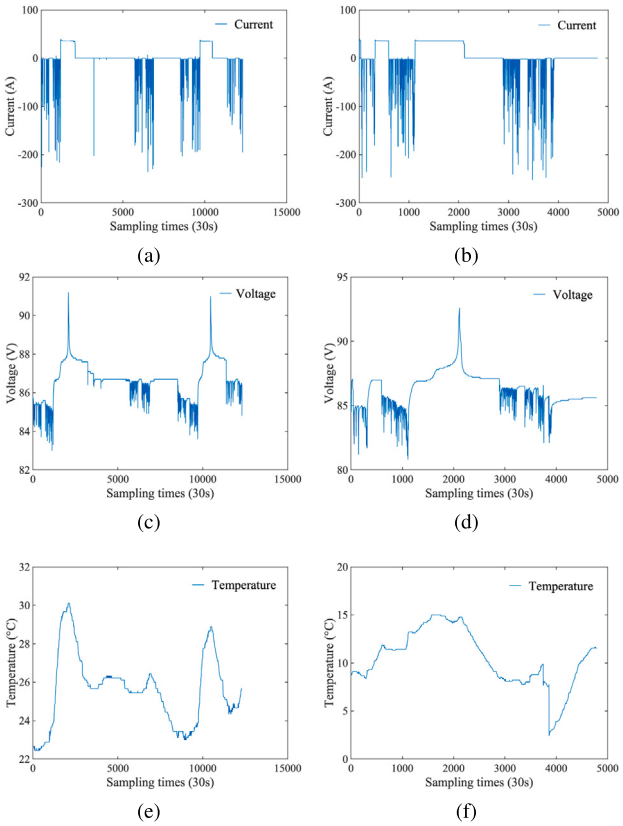


Fig. 7. Operating conditions of Pack 1. (a), (c), and (e) are current, voltage, and temperature in April 2023, respectively, and (b), (d), and (f) are current, voltage, and temperature in December 2022, respectively.

3.3. SOC estimation with the proposed hybrid model in electric vehicle battery packs

In this subsection, the operating data of two Electric Aerial Lift Vehicles' (EALVs) battery packs (called Pack 1 and Pack 2) were collected to verify the effectiveness of the proposed hybrid model, as shown in Fig. 6. The primary purpose is to evaluate whether the proposed hybrid model can dynamically optimize hyperparameters across diverse datasets. The two EALV models are G01BB03 and G01JB01, and the operating conditions of the two battery packs at two temperatures in April 2023 and December 2022 are illustrated in Figs. 7 and 8, respectively. Note that, for each pack, all collected data were divided into the training and test sets with a 2:1 ratio.

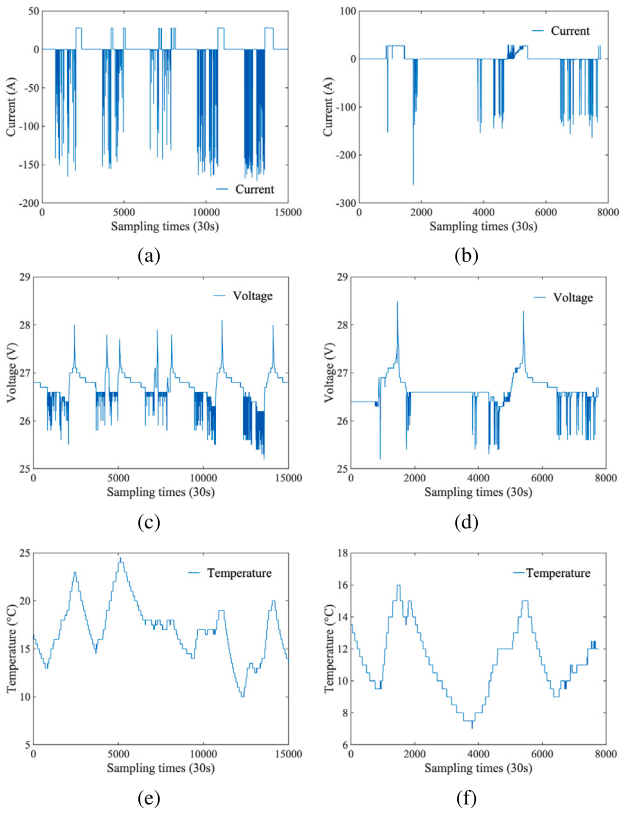


Fig. 8. Operating conditions of Pack 2. (a), (c), and (e) are current, voltage, and temperature in April 2023, respectively, and (b), (d), and (f) are current, voltage, and temperature in December 2022, respectively.

Table 4
SOC estimation results for Pack 1.

Pack 1	RMSE(%)			MAX(%)		
	GRU	GRU+ UKF	Proposed model	GRU	GRU+ UKF	Proposed model
April 2023	3.68	0.72	**0.33**	11.97	3.72	**1.37**
December 2022	1.98	0.69	**0.37**	9.31	2.67	**1.22**

Experiments were conducted on Packs 1 and 2, and the RMSE and MAX values were calculated. The results on Pack 1 are shown in Fig. 9 and Table 4, and the results on Pack 2 are shown in Fig. 10 and Table 5. Under the operating conditions of Pack 1, the proposed hybrid model outperforms the GRU network and the GRU+UKF model in terms of both RMSE and MAX. At the normal temperature of April 2023, the proposed hybrid model achieves the RMSE value of 0.33%, which is better than 3.68% of the GRU network and 0.72% of the GRU+UKF model. In addition, the proposed hybrid model reduces the MAX value from 11.97% to 1.37%. At the low temperature of December 2022, the proposed hybrid model decreases the RMSE value from 1.98% to 0.37%, and the MAX value from 9.31% to 1.22%, respectively. Under the operating conditions of Pack 2, we can draw the same conclusion that the performance of the proposed hybrid model is better than that of the GRU network and the GRU+UKF model in terms of both RMSE and MAX. The proposed hybrid model presents the RMSE values of 0.26% and 0.21% at the normal and low temperatures, respectively, and the MAX values of 1.93% and 1.49% at the normal and low temperatures, respectively. Moreover, from Fig. 9 and Fig. 10, it can be observed that the errors provided by the proposed hybrid model are consistently smaller than those derived from the GRU network and the GRU+UKF model at different sampling times.

From the above results, it is evident that the proposed hybrid model performs well in estimating the SOC of actual electric vehicle battery

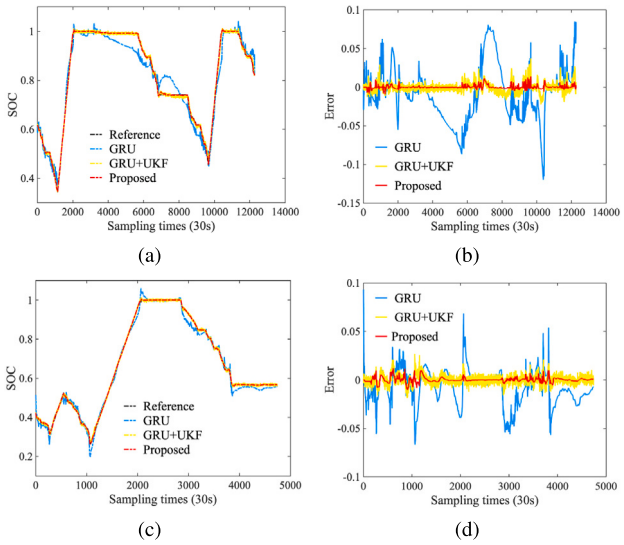


Fig. 9. Results and errors for Pack 1 in: (a) April 2023; (b) errors of (a); (c) December 2022; (d) errors of (c).

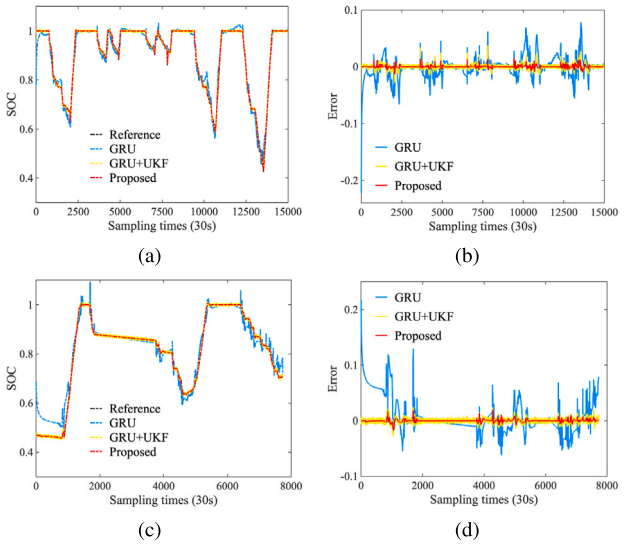


Fig. 10. Results and errors for Pack 2 in: (a) April 2023; (b) errors of (a); (c) December 2022; (d) errors of (c).

Table 5
SOC estimation results for Pack 2.

Pack 2	RMSE(%)			MAX(%)		
	GRU	GRU+ UKF	Proposed model	GRU	GRU+ UKF	Proposed model
April 2023	3.29	0.51	**0.26**	21.75	2.72	**1.93**
December 2022	1.88	0.55	**0.21**	22.22	3.75	**1.49**

packs, which demonstrates its ability to adaptively adjust network hyperparameters under varying operating conditions. The good performance of the proposed hybrid model also verifies its potential in practical applications.

3.4. Ablation study

An ablation study was conducted to validate the contribution of each component of the proposed hybrid model. The results are presented in Table 6. In this study, BL represents the baseline model, which

Table 6
Ablation study for FUDS at 0 °C, 20 °C, 30 °C, and 50 °C.

FUDS	RMSE(%)					MAX(%)				
	BL	BL+(1)	BL+(2)	BL+(3)	BL+(1)+(2)+(3)	BL	BL+(1)	BL+(2)	BL+(3)	BL+(1)+(2)+(3)
0 °C	5.71	4.17	2.63	1.87	**0.41**	19.66	11.52	8.79	4.29	**2.02**
20 °C	5.98	4.07	2.79	1.78	**0.39**	24.17	16.97	10.15	4.83	**1.63**
30 °C	6.84	5.00	3.19	2.03	**0.42**	24.59	19.17	10.91	5.27	**2.06**
50 °C	6.36	4.95	3.07	1.91	**0.39**	22.47	16.50	8.33	4.72	**1.99**

Table 7
SOC estimation results of the proposed hybrid model and other advanced hybrid models for FUDS at 20 °C.

Models	RMSE(%)	MAX(%)	Parameter optimization
LSTM-EKF [41]	0.72	2.39	No
LSTM-ACKF [42]	0.90	2.70	No
LSTM-IPF [43]	0.63	**1.15**	Yes
RC-PSO-TCN [44]	1.28	4.61	Yes
Proposed model	**0.39**	1.63	Yes

refers to the original GRU network without optimization. In addition, (1), (2), and (3) represent the memory enhancement unit, GP-CMAES, and AUKF, respectively.

Table 6 shows that all the three components in the proposed hybrid model contribute to the improvement of SOC estimation performance. Specifically, under FUDS at 20 °C, for the RMSE metric, BL+(1) (4.07%), BL+(2) (2.79%), and BL+(3) (1.78%) are better than BL (5.98%) by 1.91%, 3.19%, and 4.20%, respectively. For the MAX metric, BL+(1) (16.97%), BL+(2) (10.15%), and BL+(3) (4.83%) are better than BL (24.17%) by 7.20%, 14.02%, and 19.34%, respectively.

From the above results, it is clear that AUKF has the most significant contribution in improving SOC estimation performance, particularly in terms of the MAX metric. However, it is important to note that while AUKF provides notable performance improvements, using AUKF alone cannot fully release the potential of the original GRU network. Therefore, by incorporating the memory enhancement unit into BL and optimizing it using GP-CMAES, the performance of the proposed hybrid model is further enhanced. For example, at 50 °C, compared with BL+(3), the RMSE and MAX metrics of BL+(1)+(2)+(3) decrease from 1.91% to 0.39% and from 4.72% to 1.99%, respectively.

3.5. Comparison with different advanced models

To evaluate the performance of the proposed hybrid model, a comparative experiment was implemented with several advanced hybrid models, including LSTM-EKF [41], LSTM-adaptive cubature Kalman filter (ACKF) [42], LSTM-improved particle filter (IPF) [43], and second order resistor-capacitance circuit-based particle swarm optimization-temporal convolutional network (RC-PSO-TCN) [44]. Table 7 presents the RMSE and MAX values of each model under FUDS at 20 °C.

It can be observed that, the RMSE values obtained by LSTM-EKF, LSTM-ACKF, LSTM-IPF, RC-PSO-TCN, and the proposed hybrid model are 0.72%, 0.90%, 0.63%, 1.28%, and 0.39%, respectively. Among these, the RMSE value of the proposed hybrid model is the best. For the MAX metric, the values provided by LSTM-EKF, LSTM-ACKF, LSTM-IPF, RC-PSO-TCN, and the proposed hybrid model are 2.39%, 2.70%, 1.15%, 4.61%, and 1.63%, respectively. The MAX value of the proposed hybrid model is the second best and slightly underperforms LSTM-IPF.

Overall, based on the above results, considering that the RMSE metric is more critical than the MAX metric in SOC estimation, we can conclude that the proposed hybrid model exhibits superior overall performance, which can be explained from the following three aspects. Firstly, the memory enhancement unit ensures the ability of the ME-GRU network to capture long-term dependencies in time-series data. Secondly, GP-CMAES automates the hyperparameter tuning process to enhance the robustness and generalization ability of the ME-GRU network. Thirdly, AUKF has strong noise suppression capabilities, which can further improve the accuracy of the ME-GRU network for SOC estimation.

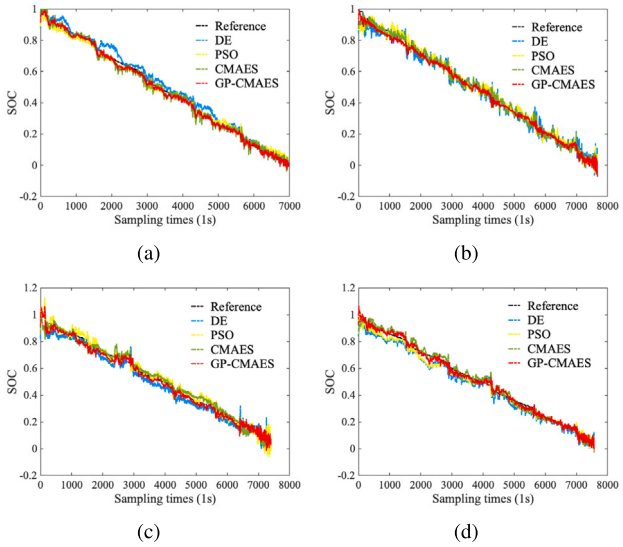


Fig. 11. Results of the proposed hybrid model with different optimization methods at: (a) 0 °C; (b) 20 °C; (c) 30 °C; (d) 50 °C.

Table 8
SOC estimation results derived from different optimization methods for FUDS at 0 °C, 20 °C, 30 °C, and 50 °C.

FUDS	RMSE(%)				MAX(%)			
	DE	PSO	CMAES	GP-CMAES	DE	PSO	CMAES	GP-CMAES
0 °C	3.23	2.42	2.17	**1.44**	10.84	9.82	8.88	**8.12**
20 °C	3.33	2.68	2.51	**1.93**	15.11	13.23	12.22	**9.73**
30 °C	4.09	3.20	3.05	**2.46**	17.62	15.30	15.91	**9.94**
50 °C	4.12	3.16	2.84	**2.01**	15.27	13.46	11.98	**7.71**

3.6. Validation of the superiority of GP-CMAES

To demonstrate the superiority of GP-CMAES in optimizing hyperparameters, we executed a comparative experiment with differential evolution algorithm (DE) [45], particle swarm optimization algorithm (PSO) [46], and CMAES under FUDS, with the results shown in Fig. 11 and Table 8. In this experiment, DE, PSO, CMAES, and GP-CMAES represent the ME-GRU network optimized by DE, PSO, CMAES, and GP-CMAES, respectively.

As shown in Table 8, GP-CMAES consistently outperforms other optimization methods across all the temperature conditions (i.e., 0 °C, 20 °C, 30 °C, and 50 °C) for both the RMSE and MAX metrics. For instance, at 50 °C, the RMSE value of GP-CMAES is 2.01%, which is superior to DE (4.12%), PSO (3.16%), and CMAES (2.84%). Regarding the MAX metric, GP-CMAES achieves 7.71%, which is lower than DE (15.27%), PSO (13.46%), and CMAES (11.98%).

These results demonstrate that, compared with existing optimization methods, GP-CMAES shows a significant advantage in hyperparameter optimization, thereby helping the ME-GRU network achieve good SOC estimation performance.

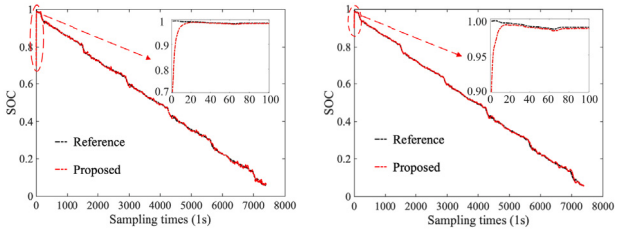


Fig. 12. Results of the proposed hybrid model under different initial SOC values: (a) 70% of the true initial value; (b) 90% of the true initial value.

3.7. Validation under different initial SOC values

In practical applications, obtaining an accurate initial SOC value is challenging due to influences such as temperature and sensor performance, which may introduce an initial error. A reliable SOC estimation method should be capable of quickly correcting the initial error. In this subsecion, we carried out an experiment to evaluate the performance of the proposed hybrid model under two different initial SOC conditions. Specifically, we set the initial SOC value to be 70% and 90% of the true initial value, respectively. The results are presented in Fig. 12. In Fig. 12, the black dashed line represents the actual SOC value, while the red dashed line represents the SOC estimation of the proposed hybrid model.

The results show that, regardless of whether the initial SOC value is 70% or 90% of the true initial value, the SOC estimation of the proposed model can gradually converge to the actual SOC value within 15 time steps. This demonstrates that even when the initial SOC value contains errors due to external factors, the proposed hybrid model still exhibits strong convergence capabilities and can estimate the SOC value accurately.

3.8. Computational efficiency

The computational efficiency of the proposed hybrid model can be divided into optimization time and computation time. The optimization time refers to the time required by GP-CMAES to determine the hyperparameters of the ME-GRU network, while the computation time refers to the time required for the proposed hybrid model to estimate the SOC of the test data. It is important to note that the optimization time is only related to the initial hyperparameter tuning process of the ME-GRU network. This implies that it does not affect the computation time or estimation speed after initialization.

We conducted a computational efficiency test under FUDS at 20 °C. The test was run based on Matlab_R2021a on an Intel machine with Core i7-10700 CPU @2.90 GHz and 16 GB RAM. The results show that the optimization time and the computation time of the proposed hybrid model are 69.19 s and 3.73 s, respectively, resulting in a total computational cost of 72.92 s.

In summary, the computational efficiency of the proposed hybrid model is relatively high, making it suitable for practical engineering applications.

4. Conclusions

In this paper, a hybrid model was proposed by combining an ME-GRU network with AUKF to achieve accurate SOC estimation of lithium-ion batteries. Firstly, the ME-GRU network produced preliminary SOC estimation results, and incorporated a memory enhancement unit to effectively exploit historical data, thereby enhancing the SOC estimation accuracy of the GRU network. In addition, AUKF corrected the preliminary SOC estimation results to mitigate the negative effect of inherent noise in the data. Moreover, GP-CMAES was developed to adaptively optimize the hyperparameters of the ME-GRU network. GP-CMAES avoided the time-consuming hyperparameter design and ensured the adaptability of the proposed hybrid model under different operating conditions.

To evaluate the effectiveness of the proposed hybrid model, battery SOC estimation was first performed in a typical scenario. The results demonstrated that the proposed hybrid model outperformed the GRU network and the GRU+UKF model, with the RMSE and MAX values below 0.60% and 2.50%, respectively. Subsequently, the performance of the proposed hybrid model was validated by a collected battery pack dataset, demonstrating its accurate SOC estimation capability at both normal and low temperatures. Specifically, in Pack 1, the RMSE and MAX values at the normal temperature were 0.33% and 1.37%, respectively, while those at the low temperature were 0.37% and 1.22%, respectively. Overall, the hybrid model proposed in this paper could achieve accurate and stable SOC estimation results in different operating conditions. Moreover, the model did not require the precise battery modeling and time-consuming hyperparameter design process.

CRediT authorship contribution statement

Wei Wang: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Yong Wang:** Writing – review & editing, Supervision, Resources, Formal analysis, Conceptualization. **Zijun Zhang:** Writing – review & editing, Supervision, Formal analysis.

Declaration of competing interest

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

Acknowledgments

This work is financially supported by the National Natural Science Foundation of China under Grant U23A20347, in part by the Fundamental Research Funds for the Central Universities of Central South University, and in part by the High Performance Computing Center of Central South University.

Data availability

Data will be made available on request.

References

[1] Shunli Wang, Chao Wang, Paul Takyi-Aninakwa, Siyu Jin, Carlos Fernandez, Qi Huang, An improved parameter identification and radial basis correction-differential support vector machine strategies for state-of-charge estimation of urban-transportation-electric-vehicle lithium-ion batteries, J. Energy Storage 80 (2024) 110222.

[2] Hany M. Hasanien, Ibrahim Alsaleh, Marcos Tostado-Véliz, Abdullah Alassaf, Ayoob Alateeq, Francisco Jurado, Optimal parameters estimation of lithium-ion battery in smart grid applications based on gazelle optimization algorithm, Energy 285 (2023) 129509.

[3] Liping Chen, Siqiang Xie, António M. Lopes, Huafeng Li, Xinyuan Bao, Chaolong Zhang, Penghua Li, A new SOH estimation method for lithium-ion batteries based on model-data-fusion, Energy 286 (2024) 129597.

[4] Yuan Cao, Jaber Abu Qahouq, Small-signal modeling and analysis for a wirelessly distributed and enabled battery energy storage system of electric vehicles, Appl. Sci. 9 (20) (2019) 4249.

[5] Xiaosong Hu, Fei Feng, Kailong Liu, Lei Zhang, Jiale Xie, Bo Liu, State estimation for advanced battery management: Key challenges and future trends, Renew. Sustain. Energy Rev. 114 (2019) 109334.

[6] Dongdong Qiao, Xueyuan Wang, Xin Lai, Yuejiu Zheng, Xuezhe Wei, Haifeng Dai, Online quantitative diagnosis of internal short circuit for lithium-ion batteries using incremental capacity method, Energy 243 (2022) 123082.

[7] Yanxin Xie, Shunli Wang, Gexiang Zhang, Yongcun Fan, Carlos Fernandez, Frede Blaabjerg, Optimized multi-hidden layer long short-term memory modeling and suboptimal fading extended Kalman filtering strategies for the synthetic state of charge estimation of lithium-ion batteries, Appl. Energy 336 (2023) 120866.
[8] Kieran Mc Carthy, Hemtej Gullapalli, Kevin M. Ryan, Tadhg Kennedy, Use of impedance spectroscopy for the estimation of li-ion battery state of charge, state of health and internal temperature, J. Electrochem. Soc. 168 (8) (2021) 080517.
[9] Quanqing Yu, Yukun Huang, Aihua Tang, Chun Wang, Weixiang Shen, OCV-SOC-temperature relationship construction and state of charge estimation for a series–parallel lithium-ion battery pack, IEEE Trans. Intell. Transp. Syst. 24 (6) (2023) 6362–6371.
[10] Jinmei Xu, Shengkun Xie, Zhen Lin, Xiangyun Qiu, Kai Wu, Honghe Zheng, Studies of interfacial reaction characteristics for high power lithium-ion battery, Electrochim. Acta 435 (2022) 141305.
[11] Jeong Lee, Jehyuk Won, Enhanced Coulomb counting method for SoC and SoH estimation based on Coulombic efficiency, IEEE Access 11 (2023) 15449–15459.
[12] Kiarash Movassagh, Arif Raihan, Balakumar Balasingam, Krishna Pattipati, A critical look at Coulomb counting approach for state of charge estimation in batteries, Energies 14 (14) (2021).
[13] Jiankun Peng, Jiayi Luo, Hongwen He, Bing Lu, An improved state of charge estimation method based on cubature Kalman filter for lithium-ion batteries, Appl. Energy 253 (2019) 113520.
[14] Xiaoqing Ren, Shulin Liu, Xiaodong Yu, Xia Dong, A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM, Energy 234 (2021) 121236.
[15] L. Ma, C. Hu, F. Cheng, State of charge and state of energy estimation for lithium-ion batteries based on a long short-term memory neural network, J. Energy Storage 37 (2021) 102440.
[16] Cong Jiang, Shunli Wang, Bin Wu, Carlos Fernandez, Xin Xiong, James Coffie-Ken, A state-of-charge estimation method of the power lithium-ion battery in complex conditions based on adaptive square root extended Kalman filter, Energy 219 (2021) 119603.
[17] Xinze Zhao, Bingxiang Sun, Weige Zhang, Xitian He, Shichang Ma, Junwei Zhang, Xiaopeng Liu, Error theory study on EKF-based SOC and effective error estimation strategy for li-ion batteries, Appl. Energy 353 (2024) 121992.
[18] Yong Tian, Bizhong Xia, Wei Sun, Zhihui Xu, Weiwei Zheng, A modified model based state of charge estimation of power lithium-ion batteries using unscented Kalman filter, J. Power Sources 270 (2014) 619–626.
[19] Rui Xiong, Jinpeng Tian, Weixiang Shen, Fengchun Sun, A novel fractional order model for state of charge estimation in lithium ion batteries, IEEE Trans. Veh. Technol. 68 (5) (2019) 4130–4139.
[20] Ignacio E. Alćantara, Angel L. Cedeño, César A. Silva, Juan C. Agüero, Li-ion battery SOC estimation via Gaussian sum filtering and piecewise linear approximation, in: 2024 IEEE ANDESCON, IEEE, 2024, pp. 1–6.
[21] Yujie Wang, Zonghai Chen, A framework for state-of-charge and remaining discharge time prediction using unscented particle filter, Appl. Energy 260 (2020) 114324.
[22] Zhenggang Chen, Jianxiong Zhou, Fei Zhou, Shuai Xu, State-of-charge estimation of lithium-ion batteries based on improved h infinity filter algorithm and its novel equalization method, J. Clean. Prod. 290 (2021) 125180.
[23] Chuang Sheng, Jun Fu, Dong Li, Chang Jiang, Ziang Guo, Beijia Li, Jingzhi Lei, Linghong Zeng, Zhonghua Deng, Xiaowei Fu, Xi Li, Energy management strategy based on health state for a PEMFC/Lithium-ion batteries hybrid power system, Energy Convers. Manage. 271 (2022) 116330.
[24] Chengqi She, Lei Zhang, Zhenpo Wang, Fengchun Sun, Peng Liu, Chunbao Song, Battery state-of-health estimation based on incremental capacity analysis method: Synthesizing from cell-level test to real-world application, IEEE J. Emerg. Sel. Top. Power Electron. 11 (1) (2023) 214–223.
[25] Bin Gou, Yan Xu, Xue Feng, State-of-health estimation and remaining-useful-life prediction for lithium-ion battery using a hybrid data-driven method, IEEE Trans. Veh. Technol. 69 (10) (2020) 10854–10867.
[26] Xinyuan Bao, Liping Chen, António M. Lopes, Xin Li, Siqiang Xie, Penghua Li, YangQuan Chen, Hybrid deep neural network with dimension attention for state-of-health estimation of lithium-ion batteries, Energy 278 (2023) 127734.
[27] Isaiah Oyewole, Abdallah Chehade, Youngki Kim, A controllable deep transfer learning network with multiple domain adaptation for battery state-of-charge estimation, Appl. Energy 312 (2022) 118726.
[28] Hicham Chaoui, Chinemerem Christopher Ibe-Ekeocha, State of charge and state of health estimation for lithium batteries using recurrent neural networks, IEEE Trans. Veh. Technol. 66 (10) (2017) 8773–8783.
[29] Shiva Nosouhian, Fereshteh Nosouhian, Abbas Kazemi Khoshouei, A review of recurrent neural network architecture for sequence learning: Comparison between LSTM and GRU, 2021, Preprints.
[30] Panagiotis Eleftheriadis, Sonia Leva, Emanuele Ogliari, Bayesian hyperparameter optimization of stacked bidirectional long short-term memory neural network for the state of charge estimation, Sustain. Energy, Grids Netw. 36 (2023) 101160.
[31] Zhen Zhang, Ming Xu, Longhua Ma, Binchao Yu, A state-of-charge estimation method based on bidirectional lstm networks for lithium-ion batteries, in: 2020 16th International Conference on Control, Automation, Robotics and Vision, ICARCV, 2020, pp. 211–216.
[32] Mahammad A. Hannan, Dickson N.T. How, Muhamad Bin Mansor, Md S. Hossain Lipu, Pin Jern Ker, Kashem M. Muttaqi, State-of-charge estimation of Li-ion battery using gated recurrent unit with one-cycle learning rate policy, IEEE Trans. Ind. Appl. 57 (3) (2021) 2964–2971.
[33] Carlos Vidal, Pawel Malysz, Mina Naguib, Ali Emadi, Phillip J Kollmeyer, Estimating battery state of charge using recurrent and non-recurrent neural networks, J. Energy Storage 47 (2022) 103660.
[34] Nikolaus Hansen, Sibylle D. Müller, Petros Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 11 (1) (2003) 1–18.
[35] Zhiyao Zhang, Yong Wang, Guangyong Sun, Tong Pang, Ke Tang, Constrained probabilistic Pareto dominance for expensive constrained multiobjective optimization problems, IEEE Trans. Evol. Comput. (2024).
[36] Zhiyao Zhang, Yong Wang, Jiao Liu, Guangyong Sun, Ke Tang, A two-phase kriging-assisted evolutionary algorithm for expensive constrained multiobjective optimization problems, IEEE Trans. Syst. Man, Cybern.: Syst. (2024).
[37] Zhiyao Zhang, Yong Wang, Guangyong Sun, Tong Pang, A distribution information-based kriging-assisted evolutionary algorithm for expensive many-objective optimization problems, IEEE Trans. Evol. Comput. (2024).
[38] Bing-Chuan Wang, Yan-Bo He, Jiao Liu, Biao Luo, Fast parameter identification of lithium-ion batteries via classification model-assisted Bayesian optimization, Energy 288 (2024) 129667.
[39] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, Thomas Stützle, The irace package: Iterated racing for automatic algorithm configuration, Oper. Res. Perspect. 3 (2016) 43–58.
[40] Zhe Jiang, Qi Song, Yuqing He, Jianda Han, A novel adaptive unscented Kalman filter for nonlinear estimation, in: 2007 46th IEEE Conference on Decision and Control, IEEE, 2007, pp. 4293–4298.
[41] Xingtao Liu, Wenlong Xia, Siyuan Li, Mingqiang Lin, Ji Wu, State of charge estimation for lithium-ion battery pack with selected representative cells, IEEE Trans. Transp. Electrification (2023).
[42] Yong Tian, Rucong Lai, Xiaoyu Li, Lijuan Xiang, Jindong Tian, A combined method for state-of-charge estimation for lithium-ion batteries using a long short-term memory network and an adaptive cubature Kalman filter, Appl. Energy 265 (2020) 114789.
[43] Xiaodong Yan, Gongbo Zhou, Wei Wang, Ping Zhou, Zhenzhi He, A hybrid data-driven method for state-of-charge estimation of lithium-ion batteries, IEEE Sensors J. 22 (16) (2022) 16263–16275.
[44] Feng Li, Wei Zuo, Kun Zhou, Qingqing Li, Yuhan Huang, Guangde Zhang, State-of-charge estimation of lithium-ion battery based on second order resistor-capacitance circuit-PSO-TCN model, Energy 289 (2024) 130025.
[45] Kenneth V. Price, Differential evolution, in: Handbook of Optimization: From Classical to Modern Approach, Springer, 2013, pp. 187–214.
[46] James Kennedy, Russell Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.