energies

Article

Collaborative Estimation of Lithium Battery State of Charge Based on the BiLSTM-AUKF Fusion Model

Rui Wang [1,2,3], Lele Liu [1], Honghou Zhang [3], Qifeng Qian [3], Lingchao Xiao [3], Qiansheng Qiu [3], Chao Tan [2] and Fujian Yang [4,*]

[1] School of Wangzheng Microelectronics, Changzhou University, Changzhou 213164, China; wangrui@cczu.edu.cn (R.W.); lle701@outlook.com (L.L.)
[2] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China; tanchao@tju.edu.cn
[3] Zhejiang Sunoren Solar Technology Co., Ltd., Haining 314400, China
[4] School of Urban Construction, Changzhou University, Changzhou 213164, China
* Correspondence: fjyang@outlook.com

Abstract

To address the issue of decreased accuracy in lithium battery state of charge (SOC) estimation caused by parameter mismatches, modeling error accumulation, and sensitivity to noise, this paper proposes a collaborative estimation method. The proposed method combines a Bayesian optimization (BO)-tuned dual-input bidirectional long short-term memory network (BiLSTM) with an adaptive unscented Kalman filter (AUKF) based on the Sage–Husa adaptive strategy. First, a dual-input BiLSTM network is constructed using a multi-layer cascaded BiLSTM to extract time-dependent features. This network fuses both temporal and static features to perform an initial SOC prediction, while BO is employed to adaptively optimize the network's hyperparameters. Second, the BiLSTM prediction outputs and the physical model are incorporated into the AUKF framework to achieve real-time iterative SOC estimation. Multi-scenario experiments conducted on the University of Maryland CALCE battery dataset demonstrated that the proposed method achieved a mean absolute error (MAE) below 0.6% and a root mean square error (RMSE) less than 0.8%. This method effectively enhances the robustness and noise immunity of SOC estimation in dynamic scenarios, providing a high-precision state estimation solution for battery management systems.

Keywords: lithium-ion battery; state of charge; Bayesian optimization; long short-term memory network; adaptive unscented Kalman filter

check for updates

Academic Editor: JongHoon Kim

Received: 15 September 2025
Revised: 18 October 2025
Accepted: 23 October 2025
Published: 26 October 2025

Citation: Wang, R.; Liu, L.; Zhang, H.; Qian, Q.; Xiao, L.; Qiu, Q.; Tan, C.; Yang, F. Collaborative Estimation of Lithium Battery State of Charge Based on the BiLSTM-AUKF Fusion Model. *Energies* **2025**, *18*, 5624. https://doi.org/10.3390/en18215624

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Accurate estimation of SOC in lithium-ion batteries serves as a core technology for battery management systems to achieve energy optimization, lifespan prediction, and safety control [1]. However, the SOC of lithium batteries cannot be directly measured. Moreover, the complex electrochemical characteristics within batteries and the highly nonlinear relationships under dynamic operating conditions result in challenges in SOC estimation, such as multi-timescale coupling, time-varying parameters, and noise interference [2]. Existing methods are primarily categorized into three approaches: direct measurement, model-driven, and data-driven techniques [3]. Direct measurement methods are exemplified by the open-circuit voltage (OCV) method and the ampere-hour (Ah) integration method. The OCV method directly maps SOC through the OCV-SOC calibration curve, but it requires

prolonged battery rest periods, making it impractical for real-time applications [4,5]. The Ah integration method calculates SOC by accumulating charge and discharge currents. However, it is susceptible to initial SOC errors and current sensor noise [6].

Model-based approaches utilize physical models combined with filtering algorithms to achieve SOC estimation, such as the Kalman filter (KF) and particle filters [7–11]. Commonly adopted models include the equivalent circuit model (ECM) or electrochemical models [12]. Although electrochemical models can fundamentally reveal complex internal reactions through physical and chemical principles, their practical application remains limited due to inherent complexity and difficulties in accurate parameter identification [13]. Among ECMs, the Thevenin model is widely adopted for its computational simplicity and ability to reflect internal mechanisms to a certain extent [14–16]. However, the model-based approach emphasizes the accuracy of the model, and the variation of model parameters in different environments and states makes it challenging to maintain the accuracy of SOC estimation.

Data-driven approaches leverage machine learning and deep learning algorithms to establish a mapping relationship between SOC and battery measurement data [17–22]. These methods yield accurate estimation results and demonstrate strong capabilities in handling nonlinear relationships [23]. However, they suffer from notable drawbacks, including a lack of explicit physical interpretability, time-consuming training processes, and performances that heavily depend on data quality [24]. In practical applications, ensuring the quality of collected data remains challenging due to sensor inaccuracies and external environmental interference.

Therefore, combining the strengths of different methods for joint estimation has become a research focus [25–27]. Yang et al. [28] employed an LSTM network to estimate battery SOC, followed by a UKF to reduce estimation errors. Xu et al. [29] utilized a broad learning system to model battery voltage characteristics and subsequently applied UKF for SOC estimation. Tian et al. [30], based on a first-order RC equivalent circuit model and an extended Kalman filter (EKF), decomposed the battery terminal voltage to extract internal physical information, which was then fed into an LSTM for training. This approach enriched the neural network's feature inputs and enhanced SOC estimation performance. Takyi-Aninakwa et al. [31] adopted an adaptive singular value decomposition-based unscented Kalman filter combined with an LSTM extended-input model. This framework reliably and accurately estimates SOC under diverse operating conditions by incorporating battery domain-specific parameters. Chen et al. [32] leveraged LSTM to establish a coarse estimation model linking input voltage, current, operating temperature, and state of health to SOC. An adaptive H-infinity Filter was then applied to suppress output fluctuations and improve estimation accuracy. Similarly, Wang et al. [33] proposed a closed-loop framework based on a deep convolutional neural network. To enhance the filter's robustness against non-Gaussian noise, a maximum correntropy square-root cubature Kalman filter was applied to smooth the network's SOC output. For further computational efficiency and stability, the literature [34] introduced a novel SOC estimation method using a simplified gated recurrent unit (GRU) structure integrated with an adaptive Kalman filter (AKF). Initial filter parameters were set based on the evolution of the Kalman gain to achieve superior SOC convergence. Additionally, Wei et al. [35] combined a nonlinear autoregressive model with exogenous inputs (NARX) with AKF, while [36] integrated Ada-CNN-GRU with KF. Consequently, compared to purely model-based or data-driven approaches, these hybrid methods exhibit enhanced observability, reduced output fluctuations, and superior SOC estimation performance.

However, these hybrid strategies still face multiple challenges in practical deployment. Most methods are not only highly sensitive to parameter tuning but also prone to error accumulation over prolonged operation. Moreover, their filters often rely on static noise

assumptions, which struggle to adapt to the dynamic changes in real-world environments. The widespread dependence on manual parameterization not only reduces practicality but also introduces the risk of subjective errors. Furthermore, co-estimation frameworks are often plagued by high computational complexity and inadequate handling of model–data mismatches—a shortcoming that is particularly exacerbated under varying temperatures and aging conditions. As a result, the stability of existing algorithms is compromised in practical applications due to their complexity and parameter sensitivity. The need for frequent manual adjustments and retraining in dynamic environments further diminishes operational efficiency and may introduce additional errors due to human intervention, leading to performance degradation.

To address these challenges, this study proposes a synergistic estimation framework that integrates a BO-tuned dual-input BiLSTM network with an AUKF. The methodology implements a three-phase optimization: (1) A BO-optimized BiLSTM fuses temporal and static features for high-precision initial SOC prediction; (2) the particle swarm optimization (PSO) identifies ECM parameters to establish physical voltage-SOC constraints; (3) BiLSTM outputs and ECM are embedded within the UKF framework, where the Sage-Husa strategy dynamically calibrates noise statistics online to suppress error accumulation. Experimental validation across various temperatures and operating conditions confirmed the algorithm's superior estimation accuracy and enhanced robustness.

2. Problem Model

The internal circuit configurations of different energy storage systems (ESS) vary significantly, posing challenges in accurately obtaining internal circuit details during SOC estimation. To address this, it is necessary to develop charge–discharge models that establish a mathematical relationship between SOC and circuit parameters. In ECMs, while the first-order RC model has a simple structure, it often fails to accurately capture dynamic characteristics such as the slow diffusion processes within the battery. In contrast, higher-order RC network models can slightly improve accuracy but require the introduction of numerous unknown parameters, leading to increased model identification complexity. Considering the optimal balance between accuracy and complexity, we adopted the widely used second-order RC equivalent circuit model, structure shown in Figure 1. The second-order RC network effectively represents key dynamic characteristics such as electrochemical polarization and concentration polarization. Furthermore, the parameters of this model can be conveniently identified experimentally, ensuring both feasibility for real-time battery management system (BMS) estimation and reliability.

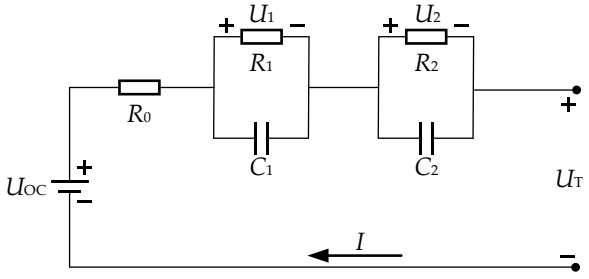


Figure 1. Second-Order RC equivalent circuit model.

In this model, the first resistor-capacitor (RC) network is employed to characterize the electrochemical polarization induced by the battery's charge transfer process, while the second RC network models the concentration polarization arising from lithium-ion

diffusion. The polarization voltages U_1 and U_2 corresponding to these two RC networks satisfy the following system of differential equations, i.e.,

$$\begin{cases} \frac{dU_1}{dt} = -\frac{U_1}{R_1C_1} + \frac{I}{C_1} \\ \frac{dU_2}{dt} = -\frac{U_2}{R_2C_2} + \frac{I}{C_2} \end{cases} \tag{1}$$

Based on Figure 1, the relationship between the voltages in the equivalent circuit can be expressed as

$$U_T = U_{OC}(SOC) - I \cdot R_0 - U_1 - U_2 \tag{2}$$

where the U_T is the terminal voltage, the U_{OC} denotes the open-circuit voltage, and the relationship between U_{OC} and SOC can be determined through low-rate battery's charge-discharge experiments, as detailed in Section 7.2.

Using the Ah integration method, the SOC variation is calculated by integrating the battery's charging or discharging current during operation, as given by

$$SOC(t) = SOC_0 - \frac{1}{C_{nom}} \int_0^t \eta I(\tau) d\tau \tag{3}$$

where SOC_0 represents the initial SOC, C_{nom} denotes the battery's available capacity, $I(t)$ is the instantaneous current, and η stands for the Coulombic efficiency ($\eta \leq 1$ during charging since part of the electrical energy is lost as heat dissipation, while typically set to $\eta = 1$ during discharging).

Discretizing Equations (1)–(3) into the discrete equivalent circuit model for SOC estimation, we obtain the corresponding discrete-time model as

$$\begin{cases} SOC(k) = SOC(k-1) - \frac{\eta \times \Delta t}{C_{nom}} I(k-1) \\ U_T(k) = U_{OC}(SOC(k)) - R_0 \times I(k) - U_1(k) - U_2(k) \\ U_1(k) = e^{-\frac{\Delta t}{R_1C_1}} U_1(k-1) + R_1(1 - e^{-\frac{\Delta t}{R_1C_1}}) I(k-1) \\ U_2(k) = e^{-\frac{\Delta t}{R_2C_2}} U_2(k-1) + R_2(1 - e^{-\frac{\Delta t}{R_2C_2}}) I(k-1) \end{cases} \tag{4}$$

Let the model parameter vector of the equivalent circuit in Equation (4) be denoted as $\boldsymbol{V} = [R_0, R_1, R_2, C_1, C_2]^T$. By minimizing the mean square error between the simulated terminal voltage $U_T(k)$ and the corresponding measured counterpart $U_{meas}(k)$, a cost function with respect to $\boldsymbol{V}$ is formulated as

$$\boldsymbol{V} = \min_V \left\{ \frac{1}{N} \sum_{k=1}^{N} [U_T(k) - U_{meas}(k)]^2 \right\} \tag{5}$$

This cost function exhibits strong nonlinearity and involves high-dimensional unknown parameters. To prevent the solution process from converging to local minima, this study employs the PSO algorithm for parameter identification, as detailed in Section 4. Following the acquisition of equivalent circuit model parameters, SOC estimation is equivalent to a prediction problem. While such problems can be addressed using Kalman filter variants or LSTM networks, Kalman-type algorithms demonstrate high sensitivity to model inaccuracies, and LSTM suffers from the absence of physics-based constraints. Consequently, this work proposes a hybrid framework integrating BiLSTM with UKF. BiLSTM compensates for UKF's model uncertainties, while BO and Sage–Husa adaptive filtering jointly regulate the hyperparameters of the BiLSTM network and the noise statistics of the UKF algorithm.

3. Collaborative Estimation of Battery SOC Based on BiLSTM-AUKF Fusion Model

To achieve high-accuracy lithium battery SOC estimation, a dual-input BiLSTM network architecture is introduced in this paper. This design employs multi-stage cascaded BiLSTM layers to extract time-dependent features, while fusing temporal characteristics with static features for preliminary SOC prediction, which is further enhanced by BO for adaptive hyperparameter tuning. Subsequently, an ECM framework is established, in which PSO algorithms identify model parameters, constructing a nonlinear mapping between terminal voltage and SOC. The outputs from both BiLSTM predictions and this physical model are then integrated into a UKF framework to enable real-time iterative SOC estimation. Furthermore, the UKF implementation incorporates a Sage-Husa adaptive strategy to dynamically calibrate state and observation noise statistics during operation. This dual approach effectively mitigates cumulative model errors while significantly enhancing noise robustness. Figure 2 depicts the comprehensive flowchart for the BiLSTM-AUKF fusion model approach to power lithium-ion battery SOC estimation.

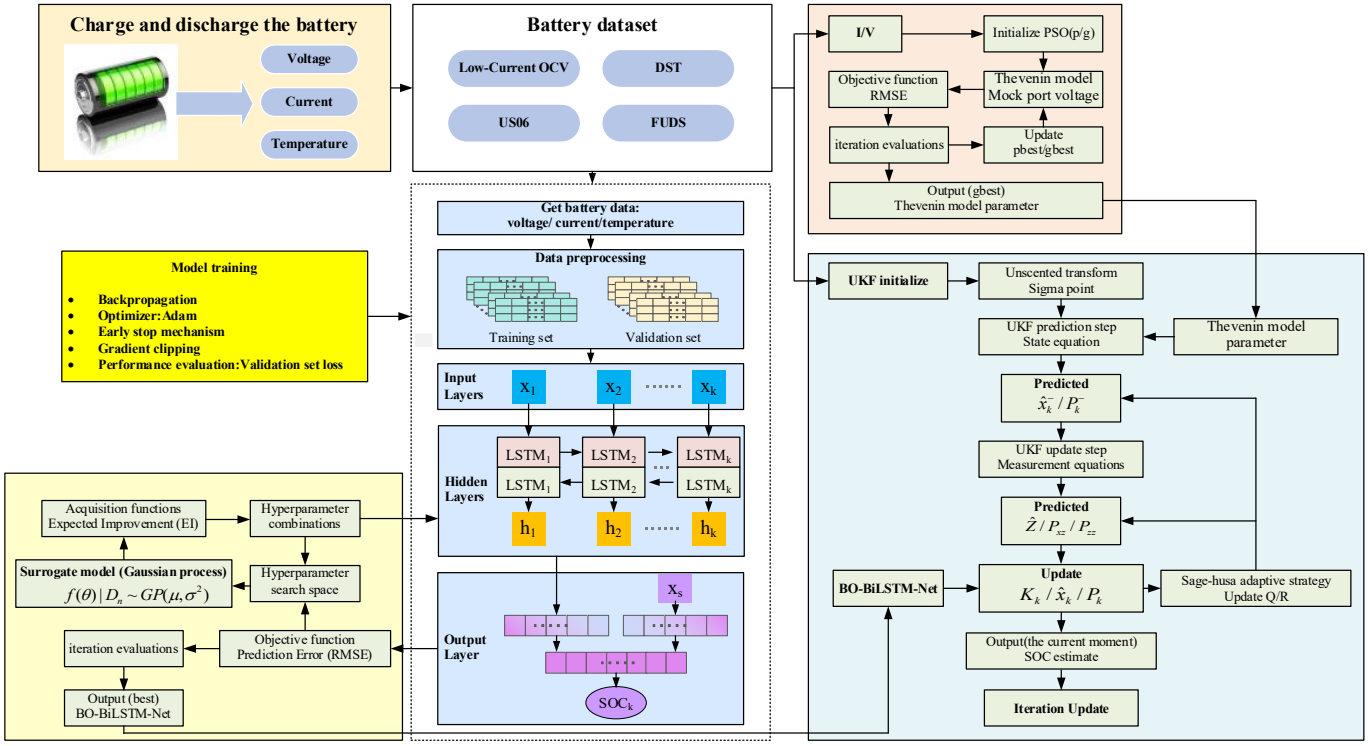


Figure 2. Flowchart of battery SOC estimation using BiLSTM-AUKF fusion model.

4. PSO-Based Parameter Identification for Battery ECM

Rooted in swarm intelligence theory, the PSO emulates collective foraging mechanisms in bird flocks. Leveraging its powerful global exploration capabilities, PSO effectively enhances the identification accuracy for multiparameter nonlinear systems. This study employs the PSO algorithm to identify the parameter vector $V = [R_0, R_1, R_2, C_1, C_2]^T$ of the equivalent circuit model, with the procedure illustrated in Figure 3. Algorithmically, each particle represents a five-dimensional parameter vector, and its trajectory characterizes the dynamic optimization process.

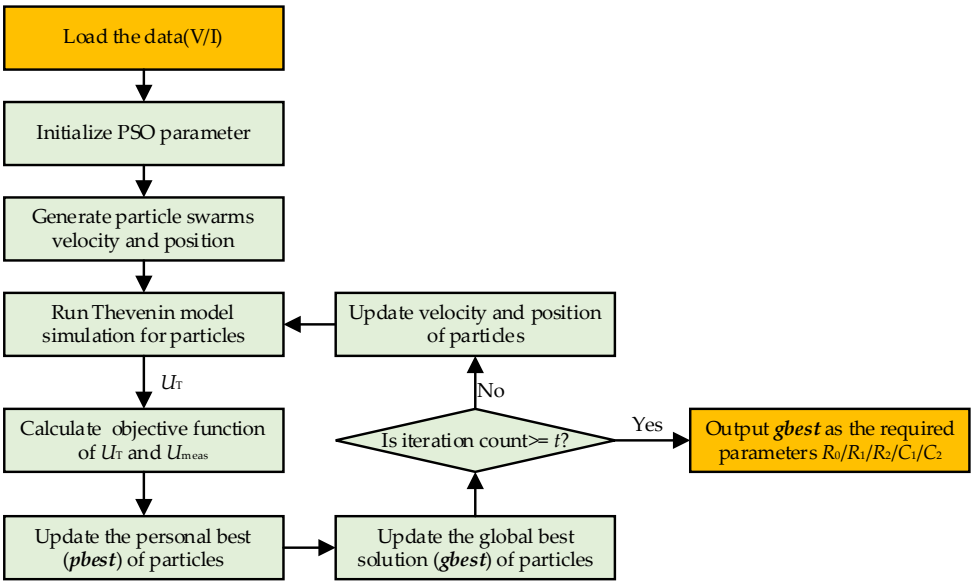


Figure 3. Flowchart of PSO-based parameter identification for battery ECM.

Within the PSO algorithm framework, given a population size of n, the position vector $\boldsymbol{P}_i = [x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}, x_{i,5}]^{\mathrm{T}}$ and velocity vector $\boldsymbol{S}_i = [s_{i,1}, s_{i,2}, s_{i,3}, s_{i,4}, s_{i,5}]^{\mathrm{T}}$ are defined for each particle i in the five-dimensional parameter space. Each dimension of $\boldsymbol{P}_i$ corresponds to an element of the equivalent circuit parameter vector $\boldsymbol{V}$ under identification, while $\boldsymbol{S}_i$ characterizes the search direction and step size of the parameter vector. The RMSE between measured terminal voltage U_{meas} and model output voltage U_{T} serves as the fitness function. Through iterative updates of particle positions and velocities, the RMSE value is minimized, driving $\boldsymbol{P}_i$ to progressively approach the true physical parameters. Each particle's position and velocity are dynamically adjusted according to its personal best ($\boldsymbol{pbest}_i$) and the global best ($\boldsymbol{gbest}$). Equation (6) defines $\boldsymbol{pbest}_i^t$ as the position corresponding to the minimum fitness value (RMSE) discovered by particle i at iteration t, and $\boldsymbol{gbest}^t$ is the position of the minimum fitness value among all particles' personal bests.

$$\begin{cases} \boldsymbol{pbest}_i^t = \mathrm{argmin}_{T\in\{1,2,\ldots,t\}}\mathrm{RMSE}(\boldsymbol{P}_i^T) \\ \boldsymbol{gbest}^t = \mathrm{argmin}_{i\in\{1,2,\ldots,n\}}\mathrm{RMSE}(\boldsymbol{pbest}_i^t) \end{cases} \tag{6}$$

The state update expressions for the velocity and position of particle i at iteration t are given by

$$\begin{cases} \boldsymbol{S}_i^{t+1} = \omega \cdot \boldsymbol{S}_i^t + \alpha_1 \cdot r_1 \cdot \left(\boldsymbol{pbest}_i^t - \boldsymbol{S}_i^t\right) + \alpha_2 \cdot r_2 \cdot \left(\boldsymbol{gbest}^t - \boldsymbol{S}_i^t\right) \\ \boldsymbol{P}_i^{t+1} = \boldsymbol{P}_i^t + \boldsymbol{S}_i^{t+1} \end{cases} \tag{7}$$

Here, the ω denotes the inertia weight, the α_1 and the α_2 are learning factors, the r_1 and the r_2 represent uniformly distributed random numbers within [0, 1]. Through iterative application of Equation (7), the algorithm computes the updated fitness value (RMSE) for each particle. If the new RMSE is lower than the previous value, the particle updates its state; otherwise, the original state is retained. After all particles' fitness values are updated, the system re-evaluates the global best solution ($\boldsymbol{gbest}$). If the fitness value of $\boldsymbol{gbest}$ improves, it is updated; otherwise, the current solution is maintained. The optimization terminates when the maximum iteration count t is reached. The globally optimal parameter set gbest is then assigned to the model parameter vector $\boldsymbol{V}$ for subsequent battery SOC estimation. By integrating both the personal best ($\boldsymbol{pbest}_i$) and global best ($\boldsymbol{gbest}$) solutions in a dual-level

selection mechanism, the optimization process ensures the continuous preservation of historically optimal solutions.

5. BO-BiLSTM-Based SOC Prediction

5.1. LSTM Cell Structure

As a variant of recurrent neural networks (RNNs), the LSTM fundamentally resolves the vanishing and exploding gradient issues in traditional RNNs during long sequence processing. Its core innovation lies in the integration of a persistent cell state and gated control mechanisms. The cell state enables continuous information flow across time steps, while the forget, input, and output gates dynamically regulate information retention, updating, and output. This architecture allows LSTM to capture long-term dependencies in sequential data, delivering significant advantages for battery SOC estimation that requires extended temporal dependency modeling.

The computational flow of an LSTM network cell at time step k proceeds as follows: The forget gate determines the proportion of information discarded from the current cell state. It generates output $\boldsymbol{f}_k$ based on the current input $\boldsymbol{u}_k$ and the previous hidden state $\boldsymbol{h}_{k-1}$.

$$\boldsymbol{f}_k = \sigma(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{k-1}, \boldsymbol{u}_k] + \boldsymbol{b}_f) \tag{8}$$

The input gate regulates which information from $\boldsymbol{u}_k$ and $\boldsymbol{h}_{k-1}$ is stored in the cell state $\boldsymbol{C}_k$. Input gate output $\boldsymbol{i}_k$ is generated by using the sigmoid activation function, and the new candidate value $\widetilde{\boldsymbol{C}}_k$ is generated using the tanh activation function for subsequent cell state updates.

$$\begin{cases} \boldsymbol{i}_k = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{k-1}, \boldsymbol{u}_k] + \boldsymbol{b}_i) \\ \widetilde{\boldsymbol{C}}_k = \tanh(\boldsymbol{W}_C \cdot [\boldsymbol{h}_{k-1}, \boldsymbol{u}_k] + \boldsymbol{b}_C) \end{cases} \tag{9}$$

Cell state propagation maintains information continuity across extended time intervals. The current cell state $\boldsymbol{C}_k$ is updated by combining the prior state $\boldsymbol{C}_{k-1}$ with new input data, i.e.,

$$\boldsymbol{C}_k = \boldsymbol{f}_k \cdot \boldsymbol{C}_{k-1} + \boldsymbol{i}_k \cdot \widetilde{\boldsymbol{C}}_k \tag{10}$$

The output gate controls which information from the updated cell state $\boldsymbol{C}_k$ is emitted as the current hidden state $\boldsymbol{h}_k$. Generate the output $\boldsymbol{O}_k$ of the output gate by using $\boldsymbol{u}_k$ and $\boldsymbol{h}_{k-1}$, and then combine it with the current moment's cell state $\boldsymbol{C}_k$ to generate $\boldsymbol{h}_k$, which is output to the next layer or the next time step.

$$\begin{cases} \boldsymbol{O}_k = \sigma(\boldsymbol{W}_O \cdot [\boldsymbol{h}_{k-1}, \boldsymbol{u}_k] + \boldsymbol{b}_O) \\ \boldsymbol{h}_k = \boldsymbol{O}_k \cdot \tanh(\boldsymbol{C}_k) \end{cases} \tag{11}$$

where $\boldsymbol{W}_f$, $\boldsymbol{W}_i$, $\boldsymbol{W}_O$, $\boldsymbol{W}_C$ are weight matrices, $\boldsymbol{b}_f$, $\boldsymbol{b}_i$, $\boldsymbol{b}_O$, $\boldsymbol{b}_C$ are bias vectors. The $\sigma(\cdot)$ is the sigmoid activation function (which compresses values to [0, 1]), with outputs approaching 1 indicating high information retention. The $\tanh(\cdot)$ is the tanh activation function (which outputs to $[-1, 1]$), encoding bipolar information.

This describes the LSTM network's process of updating the cell state and output (hidden state) through sequential inputs. It filters out irrelevant information while propagating critical features, thereby endowing the model with robust long-term memory retention. This architecture effectively overcomes long-distance dependency problems, enabling efficient prediction of time-series data.

5.2. BiLSTM Network

This study addresses the unidirectional limitation of LSTM networks in time-series learning, which neglects bidirectional relationships between information units and under-

utilizes features across the entire temporal sequence. By incorporating reverse-sequence correlations alongside forward dependencies, we construct a BiLSTM neural network model. The BiLSTM architecture integrates two independent LSTM networks with symmetric cell structures, identical inputs, and opposing information flow directions. This design enables simultaneous extraction of intrinsic relationships between current sequence information, historical context, and future states. Hidden-layer neurons store features from both directional sequences, significantly enhancing prediction capability and data utilization efficiency. Figure 4 illustrates the information propagation mechanism within BiLSTM neural units at each time step.

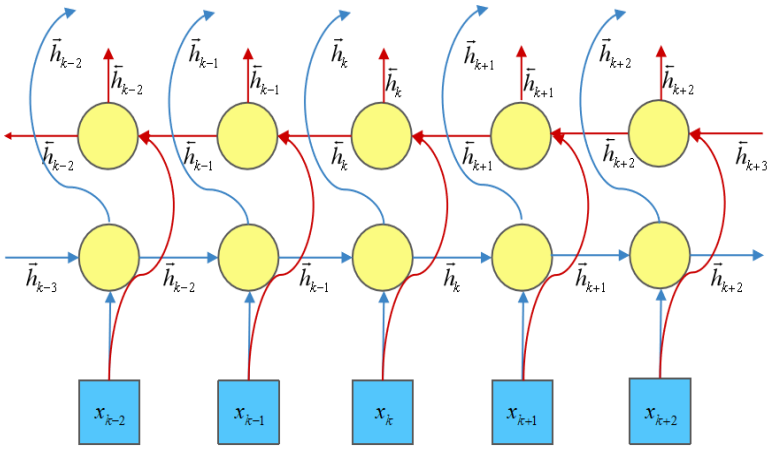


Figure 4. Schematic diagram of BiLSTM information propagation.

Take the output $\boldsymbol{h}_k{}^l$ of the lth (l = 1, 2, …, L) layer at time instant k in the hidden layer as an example.

$$\begin{cases} \overrightarrow{\boldsymbol{h}}_k = NE\left(\boldsymbol{u}_k, \overrightarrow{\boldsymbol{h}}_{k-1}\right) \\ \overleftarrow{\boldsymbol{h}}_k = NE\left(\boldsymbol{u}_k, \overleftarrow{\boldsymbol{h}}_{k+1}\right) \\ \boldsymbol{h}_k^l = \overrightarrow{\boldsymbol{h}}_k \oplus \overleftarrow{\boldsymbol{h}}_k \end{cases} \tag{12}$$

Here, $NE(\cdot)$ denotes the LSTM cell's update computation process generating outputs based on timestep inputs and the previous hidden state, with detailed equations provided in (8)–(11). The terms $\overrightarrow{\boldsymbol{h}}_k$ and $\overleftarrow{\boldsymbol{h}}_k$ represent forward-updated and backward-updated hidden states, respectively. $\oplus$ is the vector concatenation operator. The fused hidden state $\boldsymbol{h}_k{}^l \in \mathrm{R}^{2C\mathrm{units}}$, where C_{units} denotes the dimensionality of a single LSTM layer's hidden state (the number of hidden layer units). By integrating outputs from both directional hidden layers, the model captures richer informative features, significantly enhancing its learning capacity and robustness.

This paper proposes a dual-input BiLSTM architecture comprising two core modules: sequential feature learning and feature fusion, as depicted in Figure 5. In the sequential feature learning module, the sequence processing branch receives time-series input data. Specifically, sequential battery data $\boldsymbol{D}_{\mathrm{seq}} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_T]$ where $\boldsymbol{d}_k = [U_k, I_k, \mathrm{Temp}_k]^{\mathrm{T}}$. Through multiple cascaded BiLSTM layers, intrinsic temporal dependencies are extracted. For the initial L-1 layers, outputs preserve complete sequence information as $\boldsymbol{H}^l = [\boldsymbol{h}_1{}^l, \boldsymbol{h}_2{}^l, \ldots, \boldsymbol{h}_T{}^l]$. The final BiLSTM layer outputs only the last timestep's hidden state $\boldsymbol{h}_{\mathrm{seq}}{}^L$ to capture global temporal evolution characteristics. In the feature fusion module, the static feature branch receives non-sequential inputs at the current timestep. These features are encoded via a fully connected layer to produce $\boldsymbol{h}_s$, which is projected into a feature

space dimensionally compatible with the sequence branch's output. The static branch output is given by

$$\boldsymbol{h}_s = \mathrm{ReLU}(\boldsymbol{W}_s \cdot x_s + \boldsymbol{b}_s) \tag{13}$$

The static feature input in this study is defined as a scalar x_s, derived from Equation (3) as a rough SOC estimate based on Coulomb counting. This input injects physically grounded prior knowledge into the model, providing initialization for the state evolution dynamics. Subsequently, the sequence branch's final output $\boldsymbol{h}_{\mathrm{seq}}{}^{L}$ and the static feature branch's output $\boldsymbol{h}_s$ are concatenated into a fused vector $\boldsymbol{h}_{\mathrm{fusion}} = [\boldsymbol{h}_{\mathrm{seq}}{}^{L}; \boldsymbol{h}_s]$. This operation synergistically integrates historical sequential dynamics from $\boldsymbol{h}_{\mathrm{seq}}{}^{L}$ and current static state information from $\boldsymbol{h}_s$. The fused feature vector then undergoes nonlinear transformation and dimensionality reduction through a multi-layer fully connected network (incorporating ReLU activations and dropout layers), yielding $\boldsymbol{h}_{\mathrm{out}}$. This design enhances nonlinear representation capacity while mitigating overfitting risks, thereby strengthening generalization capability. The computation of $\boldsymbol{h}_{\mathrm{out}}$ is given by

$$\boldsymbol{h}_{\mathrm{out}} = \mathrm{Dropout}[\mathrm{ReLU}(\boldsymbol{W}_{\mathrm{fusion}} \cdot \boldsymbol{h}_{\mathrm{fusion}} + \boldsymbol{b}_{\mathrm{fusion}}), p] \tag{14}$$

Therefore, the predicted SOC values $\widetilde{Y}_i$ generated by forward propagation of $\boldsymbol{D}_{\mathrm{seq}}$ through the dual-input BiLSTM model are obtained as

$$\widetilde{Y}_i = \boldsymbol{W}_{\mathrm{out}} \cdot \boldsymbol{h}_{\mathrm{out}} + b_{\mathrm{out}} \tag{15}$$

Here, $\boldsymbol{D}_{\mathrm{seq}} \in R^{C \times T}$ denotes the input sequence matrix, $\boldsymbol{h}_{\mathrm{seq}}{}^{L} \in \mathrm{R}^{2\mathrm{Cunit}}$ represents the final BiLSTM hidden state. The parameter C signifies the input feature dimension, while T corresponds to the time step length. The U_k, I_k, and Temp_k represent the voltage, current, and temperature measurements at time step k in the battery time-series data. Thus, the feature dimension $C = 3$. In Equations (13)–(15), $\boldsymbol{W}_s$, $\boldsymbol{W}_{\mathrm{fusion}}$, and $\boldsymbol{W}_{\mathrm{out}}$ are weight matrices, and $\boldsymbol{b}_s$, $\boldsymbol{b}_{\mathrm{fusion}}$, and b_{out} are bias vectors. The ReLU(·) serves as the activation function, and Dropout(·) as the random masking mechanism, and p as the dropout probability.

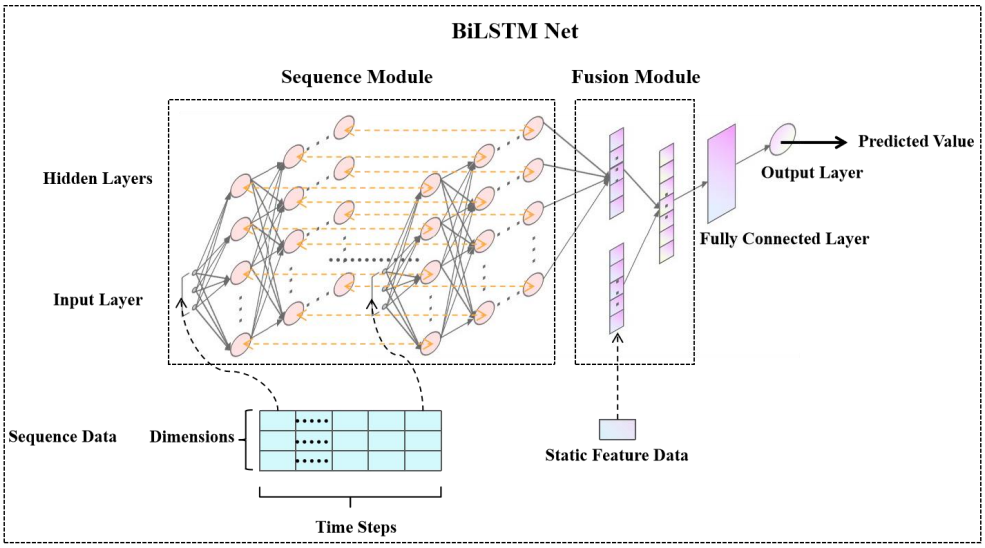


Figure 5. Architecture of dual-input BiLSTM network.

During model training, the mean squared error (MSE) serves as the regression loss function to quantify the discrepancy between the BiLSTM-predicted SOC values $\widetilde{Y}_i$ and the reference SOC values Y_i. Given a batch size B, the MSE is formally defined as

$$\varepsilon = \frac{1}{B}\sum_{i=1}^{B}\left(Y_i - \widetilde{Y}_i\right)^2 \tag{16}$$

Let the set of learnable parameters be denoted as $\boldsymbol{\varphi} = \{\boldsymbol{\varphi}_{\mathrm{BiLSTM}}, \boldsymbol{\varphi}_{\mathrm{FC}}\}$, where gradients of the loss function with respect to these parameters are computed layer-wise via backpropagation:

$$\nabla\varepsilon = \frac{\partial\varepsilon}{\partial\widetilde{\boldsymbol{Y}}} \cdot \frac{\partial\widetilde{\boldsymbol{Y}}}{\partial\boldsymbol{\varphi}} \tag{17}$$

Gradients for fully connected layers are computed layer-wise via the chain rule, while the BiLSTM layer employs the Backpropagation Through Time (BPTT) algorithm to calculate gradients accumulated across timesteps, i.e.,

$$\frac{\partial\varepsilon}{\partial\varphi_{\mathrm{BiLSTM}}} = \sum_{k=1}^{T}\frac{\partial\varepsilon}{\partial\boldsymbol{h}_{\mathrm{seq}}^{L}} \cdot \frac{\partial\boldsymbol{h}_{\mathrm{seq}}^{L}}{\partial\varphi_{\mathrm{BiLSTM}}} \tag{18}$$

The gradient of the loss function, serving as the error signal, propagates backward through the network topology via the backpropagation algorithm, traversing both fully connected layers and BiLSTM layers. The Adam gradient descent algorithm synchronously updates all learnable parameters across the sequence feature module and the feature fusion module as follows:

$$\boldsymbol{\varphi}_{\mathrm{new}} = \boldsymbol{\varphi} - \eta_L\nabla\varepsilon \tag{19}$$

Here, $\boldsymbol{\varphi}_{\mathrm{FC}}$ denotes the parameters of the feature fusion module (including the weights and biases of the static encoding layer, the concatenation layer, and the subsequent fully connected networks), $\boldsymbol{\varphi}_{\mathrm{BiLSTM}}$ represents the parameters of the sequence feature module (the weights and biases of BiLSTM layers), and η_L denotes the learning rate.

Through this architecture, end-to-end joint training is performed until model convergence, significantly enhancing the robustness and accuracy of SOC estimation under complex operating conditions and providing a reliable solution for battery state estimation.

5.3. Bayesian Optimization

The dual-input BiLSTM network architecture proposed in this study incorporates several critical hyperparameters (e.g., the number of hidden layers, the number of hidden units per layer, dropout rate, the dimensions of the fully connected layer, and the regularization coefficients). To efficiently determine the optimal configuration of these hyperparameters while avoiding laborious and subjective manual tuning, this paper employs BO for automated hyperparameter optimization. The BO algorithm minimizes the RMSE of model predictions on an independent validation set as its objective function, conducting adaptive sampling and evaluation within predefined hyperparameter search spaces.

Let the set of hyperparameters to be optimized be denoted as $\boldsymbol{\theta} \in \boldsymbol{\Psi}$, where $\boldsymbol{\theta}$ is a vector comprising the number of hidden units, dropout rate, fully connected layer dimensionality, and regularization coefficient. Here, $\boldsymbol{\Psi}$ represents the predefined feasible search space for each hyperparameter. The objective function $f(\boldsymbol{\theta})$ is defined as the RMSE of predictions generated by the BiLSTM model, trained under hyperparameter configuration $\boldsymbol{\theta}$, computed on an independent validation set. This function is minimized via Bayesian Optimization.

$$\boldsymbol{\theta}_{\mathrm{best}} = \underset{\theta\in\Psi}{\mathbf{argmin}} f(\theta) \tag{20}$$

The BO algorithm assumes that the objective function $f(\boldsymbol{\theta})$ is governed by a Gaussian Process (GP) prior distribution:

$$f(\boldsymbol{\theta}) \sim \mathcal{G}(\mu(\boldsymbol{\theta}),\, k(\boldsymbol{\theta}, \boldsymbol{\theta\prime})) \tag{21}$$

Here, $\mu(\boldsymbol{\theta})$ denotes the mean function, $k(\boldsymbol{\theta}, \boldsymbol{\theta'})$ is the covariance function (kernel function), which quantifies the correlation between any two points $\boldsymbol{\theta}$ and $\boldsymbol{\theta'}$ within the hyperparameter search space $\boldsymbol{\Psi}$. Specifically,

$$k(\boldsymbol{\theta}, \boldsymbol{\theta\prime}) = \sigma_f^2 \cdot e^{\left(-\frac{1}{2l^2}\|\boldsymbol{\theta}-\boldsymbol{\theta\prime}\|^2\right)} \tag{22}$$

The kernel function $k(\boldsymbol{\theta}, \boldsymbol{\theta'})$ quantifies the similarity in the impact of different hyperparameter configurations on model performance. When $k(\boldsymbol{\theta}, \boldsymbol{\theta'})$ approaches σ_f^2, it indicates that the two hyperparameter configurations exert strongly correlated effects on BiLSTM performance. Conversely, when $k(\boldsymbol{\theta}, \boldsymbol{\theta'})$ approaches 0, the configurations exhibit nearly independent influences on BiLSTM performance. Here, σ_f^2 denotes the signal variance, which controls the magnitude of variation in the objective function and corresponds to potential fluctuations. The l represents the length-scale, which governs the smoothness of the function.

Given that n evaluations have been conducted, the hyperparameter configurations and their corresponding objective function values constitute the observed dataset $\boldsymbol{D}_n$:

$$\boldsymbol{D}_n = \{(\boldsymbol{\theta}_i, y_i)\}_{i=1}^n \tag{23}$$

where y_i denotes the observed objective function value (validation set RMSE) obtained from the BiLSTM model under hyperparameter configuration $\boldsymbol{\theta}_i$. To account for potential stochasticity in the evaluation process, y_i is modeled as the true objective function value $f(\boldsymbol{\theta}_i)$ perturbed by Gaussian noise δ_i:

$$y_i = f(\boldsymbol{\theta}_i) + \delta_i, \delta_i \sim (0, \sigma_r^2)\Big) \tag{24}$$

where σ_r^2 denotes the noise variance, which quantifies the magnitude of evaluation errors (e.g., the fluctuations induced by training stochasticity).

Given the observed dataset $\boldsymbol{D}_n$ and GP prior, the posterior distribution of the true objective function $f(\boldsymbol{\theta^*})$ at any candidate point $\boldsymbol{\theta^*} \in \boldsymbol{\Psi}$, follows a Gaussian distribution:

$$f(\boldsymbol{\theta}^*)|\boldsymbol{D}_n \sim \mathcal{N}\Big(\mu(\boldsymbol{\theta}^*),\, \sigma^2(\boldsymbol{\theta}^*)) \tag{25}$$

The mean $\mu(\boldsymbol{\theta^*})$ and variance $\sigma^2(\boldsymbol{\theta^*})$ of the posterior distribution are given by

$$\begin{cases} \mu(\boldsymbol{\theta}^*) = \boldsymbol{G}^T\left(\boldsymbol{A} + \sigma_r^2\boldsymbol{I}\right)^{-1}\boldsymbol{y} \\ \sigma^2(\boldsymbol{\theta}^*) = k(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) - \boldsymbol{G}^T\left(\boldsymbol{A} + \sigma_r^2\boldsymbol{I}\right)^{-1}\boldsymbol{G} \end{cases} \tag{26}$$

where $\mu(\boldsymbol{\theta^*})$ represents the optimal estimate (predicted value) of $f(\boldsymbol{\theta^*})$ given the dataset $\boldsymbol{D}_n$, while $\sigma^2(\boldsymbol{\theta^*})$ quantifies the uncertainty in this estimate. The $n \times n$ covariance matrix $\boldsymbol{A}$ has elements $\boldsymbol{A}_{ij} = [k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)]$ denoting pairwise covariances between observed points. The vector $\boldsymbol{G} = [k(\boldsymbol{\theta^*}, \boldsymbol{\theta}_1), \ldots, k(\boldsymbol{\theta^*}, \boldsymbol{\theta}_n)]^T$ represents covariances between the candidate point $\boldsymbol{\theta^*}$ and observed points. The vector $y = [y_1, \ldots, y_n]^T$ contains known objective function values, and $\boldsymbol{I}$ denotes the $n \times n$ identity matrix.

The GP provides predictions (mean) and uncertainty quantification (variance) for the objective function at unevaluated points $\boldsymbol{\theta^*}$. Leveraging the posterior distribution, an acquisition function selects the next evaluation point $\boldsymbol{\theta}_{new}$ with the highest potential to improve upon the current best result. This function strategically balances exploration (sampling high-uncertainty regions) and exploitation (sampling near currently optimal

regions). This study employs expected improvement (EI) as the acquisition function, denoted $\beta_{EI}(\boldsymbol{\theta}^*)$. EI quantifies the expected improvement achievable by evaluating the objective at $\boldsymbol{\theta}^*$ relative to the current best observed value f_n^+:

$$\beta_{EI}(\boldsymbol{\theta}^*) = E[(\max(0, f(\boldsymbol{\theta}^*) - f_n^+)] \tag{27}$$

The next evaluation point $\boldsymbol{\theta}_{new}$ is selected by maximizing the EI acquisition function, i.e.,

$$\boldsymbol{\theta}_{new} = \underset{\boldsymbol{\theta}^* \in \Psi}{\mathbf{argmax}}\, \beta_{EI}(\boldsymbol{\theta}^*) \tag{28}$$

Train the BiLSTM network using the hyperparameter configuration $\boldsymbol{\theta}_{new}$, compute its RMSE on the validation set (Equation (24)) to obtain y_{new}. Augment the dataset $\boldsymbol{D}_n$ with this new observation by incorporating ($\boldsymbol{\theta}_{new}$, y_{new}), and update the Gaussian process model accordingly.

Iterate this process continuously until reaching the predefined maximum evaluation count m. Upon termination, select the optimal hyperparameter configuration $\boldsymbol{\theta}_{best}$ from all evaluated points to configure the BiLSTM network for subsequent battery SOC estimation. The Bayesian optimization workflow for the BiLSTM network is illustrated in Figure 6.

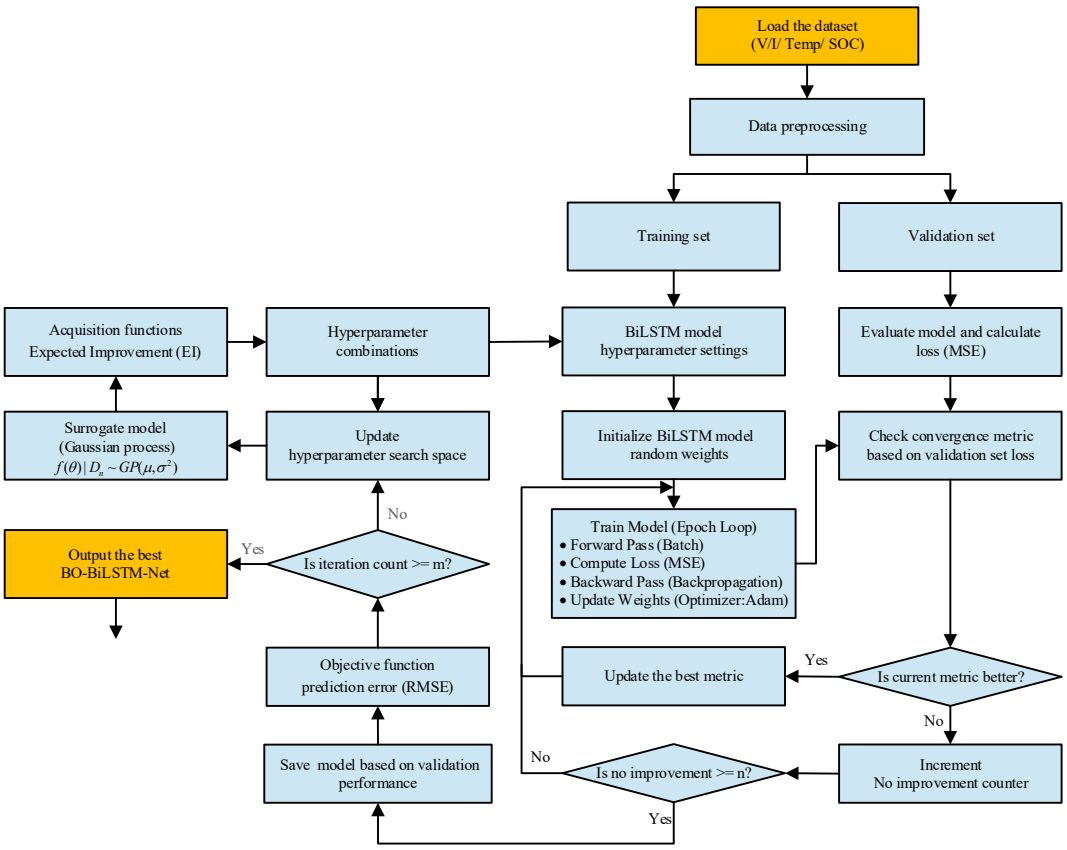


Figure 6. BO workflow for the BiLSTM model.

6. UKF with Sage-Husa Adaptive Strategy

The final framework feeds the predicted battery SOC estimates from the BO-BiLSTM network into a physics-based AUKF module. The AUKF algorithm further optimizes the SOC estimates to enhance estimation accuracy.

The physics-based AUKF defines the state vector as $\boldsymbol{x}$ = [SOC, U_1, U_2] according to Equation (4), with a dimension of n = 3. At time step k, it initializes mean weights $\boldsymbol{W}_m$ and covariance weights $\boldsymbol{W}_c$. These weights aggregate sigma points to compute statistical

quantities (mean and covariance) during the prediction and update steps. Here, α and β are scaling parameters for the unscented transform (UT), while λ denotes a composite scaling coefficient.

$$\begin{cases} \boldsymbol{W}_m[0] = \frac{\lambda}{n+\lambda}, \boldsymbol{W}_c[0] = \frac{\lambda}{n+\lambda} + \left(1 - \alpha^2 + \beta\right) \\ \boldsymbol{W}_m[i] = \boldsymbol{W}_c[i] = \frac{1}{2(n+\lambda)}, i = 1, \ldots, 2n \end{cases} \tag{29}$$

Based on the state vector $\hat{\boldsymbol{x}}_{k-1}$ and covariance matrix $\boldsymbol{P}_{k-1}$ at the previous time step, a set of sigma points is generated by using the UT to select $2n + 1$ points around $\hat{\boldsymbol{x}}_{k-1}$.

$$\begin{cases} \varsigma_{k-1}^0 = \hat{\boldsymbol{x}}_{k-1} \\ \varsigma_{k-1}^i = \hat{\boldsymbol{x}}_{k-1} + \sqrt{(n+\lambda)\boldsymbol{P}_{k-1}}, i = 1, \ldots, n \\ \varsigma_{k-1}^{n+i} = \hat{\boldsymbol{x}}_{k-1} - \sqrt{(n+\lambda)\boldsymbol{P}_{k-1}}, i = 1, \ldots, n \end{cases} \tag{30}$$

Propagate the sigma points through the state transition function $F(\cdot)$ to obtain the predicted state mean $\hat{\boldsymbol{x}}_k^-$ and predicted covariance matrix $\boldsymbol{P}_k^-$ at the current time step. The state equation $F(\cdot)$ is constructed from the state variables according to Equation (4).

$$\begin{cases} \hat{\boldsymbol{x}}_k^- = \sum_{i=0}^{2n} \boldsymbol{W}_m[i] \cdot F(\varsigma_{k-1}^i) \\ \boldsymbol{P}_k^- = \sum_{i=0}^{2n} \boldsymbol{W}_c[i] \cdot [F(\varsigma_{k-1}^i) - \hat{\boldsymbol{x}}_k^-] \cdot [F(\varsigma_{k-1}^i) - \hat{\boldsymbol{x}}_k^-]^T + \boldsymbol{Q}_{k-1} \end{cases} \tag{31}$$

Based on the state mean $\hat{\boldsymbol{x}}_k^-$ and covariance matrix $\boldsymbol{P}_k^-$ from the Prediction step, a set of sigma points is generated by using the UT to select $2n + 1$ points around $\hat{\boldsymbol{x}}_k^-$.

$$\begin{cases} \varsigma_k^0 = \hat{\boldsymbol{x}}_k^- \\ \varsigma_k^i = \hat{\boldsymbol{x}}_k^- + \sqrt{(n+\lambda)\boldsymbol{P}_k^-}, i = 1, \ldots, n \\ \varsigma_k^{n+i} = \hat{\boldsymbol{x}}_k^- - \sqrt{(n+\lambda)\boldsymbol{P}_k^-}, i = 1, \ldots, n \end{cases} \tag{32}$$

Propagate the sigma points obtained from the prediction step through the observation equation $J(\cdot)$ to compute the observation mean vector $\hat{\boldsymbol{Z}}$, observation covariance matrix $\boldsymbol{P}_{zz}$, and cross-covariance matrix $\boldsymbol{P}_{xz}$. The observation function $J(\cdot)$ is defined by Equation (2), which models battery terminal voltage.

$$\begin{cases} \hat{\boldsymbol{Z}} = \sum_{i=0}^{2n} \boldsymbol{W}_m[i] \cdot J(\varsigma_k^i) \\ \boldsymbol{P}_{xz} = \sum_{i=0}^{2n} \boldsymbol{W}_c[i] \cdot [F(\varsigma_{k-1}^i) - \hat{\boldsymbol{x}}_k^-] \cdot [J(\varsigma_k^i) - \hat{\boldsymbol{Z}}]^T \\ \boldsymbol{P}_{zz} = \sum_{i=0}^{2n} \boldsymbol{W}_c[i] \cdot [J(\varsigma_k^i) - \hat{\boldsymbol{Z}}] \cdot [J(\varsigma_k^i) - \hat{\boldsymbol{Z}}]^T + \boldsymbol{R}_{k-1} \end{cases} \tag{33}$$

The update residual $\boldsymbol{e}_k$ is computed using the actual measured terminal voltage U_{meas} and the output value $\widetilde{Y}$ from the BiLSTM network.

$$\boldsymbol{e}_k = [U_{\text{meas}}(k) - U(k); \widetilde{Y}(k) - \text{SOC}(k)] \tag{34}$$

The residual $\boldsymbol{e}_k$ is utilized to update the predicted state, yielding the updated state estimate mean vector $\hat{\boldsymbol{x}}_k$ and covariance matrix $\boldsymbol{P}_k$, thereby completing the state update and estimation for time step k. Specifically,

$$\begin{cases} \boldsymbol{K}_k = \boldsymbol{P}_{xz} \cdot \boldsymbol{P}_{zz}^{-1} \\ \hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \cdot \boldsymbol{e}_k \\ \boldsymbol{P}_k = \boldsymbol{P}_k^- - \boldsymbol{K}_k \boldsymbol{P}_{zz} \boldsymbol{K}_k{}^T \end{cases} \tag{35}$$

where $\boldsymbol{K}_k$ is the Kalman gain matrix at time step k.

In UKF, the system process noise covariance matrix $\boldsymbol{Q}_k$ and measurement noise covariance matrix $\boldsymbol{R}_k$ must be specified. Incorrect specification of $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ introduces estimation errors that accumulate over time, leading to rapid error propagation and potential filter divergence. This paper employs the Sage–Husa adaptive strategy to dynamically estimate $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$, enabling real-time adaptation to noise statistical characteristics and mitigating filter divergence. Based on the maximum likelihood principle, this strategy iteratively estimates noise statistics using filtered values and one-step predictions. After the UKF update (Equation (35)), $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are updated as follows:

$$\begin{cases} \boldsymbol{S}_k = \boldsymbol{e}_k \cdot \boldsymbol{e}_k^T - \sum_{i=0}^{2n} \boldsymbol{W}_c[i] \cdot [J(\varsigma_k^i) - \hat{\boldsymbol{Z}}] \cdot [J(\varsigma_k^i) - \hat{\boldsymbol{Z}}]^T \\ \boldsymbol{T}_k = \sum_{i=0}^{2n} \boldsymbol{W}_c[i] \cdot [F(\varsigma_{k-1}^i) - \hat{\boldsymbol{x}}_k^-] \cdot [F(\varsigma_{k-1}^i) - \hat{\boldsymbol{x}}_k^-]^T \\ \boldsymbol{R}_k = (1 - d_k) \cdot \boldsymbol{R}_{k-1} + d_k \cdot \boldsymbol{S}_k \\ \boldsymbol{Q}_k = (1 - d_k) \cdot \boldsymbol{Q}_{k-1} + d_k \cdot (\boldsymbol{K}_k \cdot \boldsymbol{e}_k \cdot \boldsymbol{e}_k^T \cdot \boldsymbol{K}_k^T + \boldsymbol{P}_k - \boldsymbol{T}_k) \end{cases} \tag{36}$$

where d_k is the forgetting factor given by $d_k = \frac{1-c}{1-c^{k+1}}$ $(0 < c < 1)$, which progressively diminishes the influence of older measurements. As time advances, d_k asymptotically approaches $1 - c$. By leveraging the Kalman gain to correct discrepancies between predicted and actual observation vectors, the state estimates are continuously iteratively updated to achieve higher-precision battery SOC estimation. The workflow is illustrated in Figure 7.

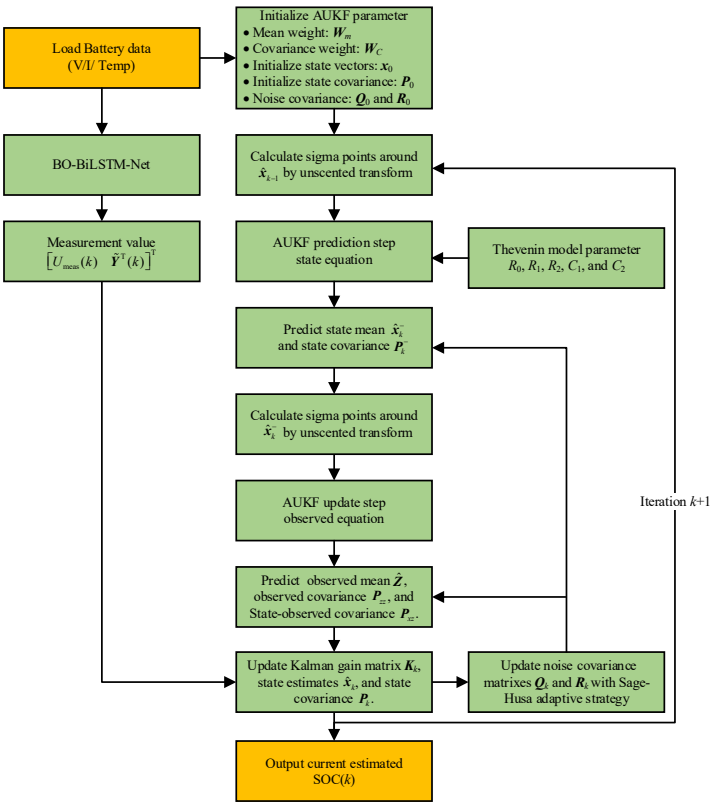


Figure 7. AUKF iterative update flowchart.

7. Model Validation and Analysis

7.1. Datasets and Metrics

To systematically validate the feasibility, accuracy, and robustness of the proposed dual-input BiLSTM-AUKF fusion model for SOC estimation in lithium-ion batteries, this study conducted experiments using an A123 $LiFePO_4$ battery with a nominal capacity of 1100 mAh, a nominal voltage of 3.3 V, and an operational voltage window between 2.0 V and 3.6 V. The hardware environment for the experiments was configured with an Intel Core i9-10900KF CPU @ 3.70 GHz, 64 GB RAM (original equipment manufacturer: Intel Corporation; headquartered in Santa Clara, CA, USA), and an NVIDIA GeForce RTX 4060 Ti GPU (NVIDIA Corporation; headquartered in Santa Clara, CA, USA), while the software platform utilized the Windows 11 operating system and MATLAB R2024a.

Initially, low-current charge–discharge cycles were performed to obtain OCV characteristic data, and the PSO algorithm was employed for parameter identification, establishing the foundation for battery modeling in subsequent SOC estimation algorithms. Subsequently, at 20 °C, the SOC estimation accuracies of GRU, LSTM, BiLSTM, BiLSTM-UKF, and the proposed dual-input BiLSTM-AUKF algorithms were compared across three typical driving cycles: dynamic stress test (DST), federal urban driving schedule (FUDS), and highway driving test (US06). By analyzing estimation results and errors, this work quantitatively evaluated the structural advantages of BiLSTM networks and the role of the Sage–Husa adaptive strategy in enhancing UKF performance. Finally, the robustness and generalization capability of the BiLSTM-AUKF algorithm were validated under inaccurate initial SOC settings (initial error) and varying ambient temperatures. First, different initial SOC values (90%, 80%, 70%; true value: 100%) were tested at 20 °C to examine their convergence robustness. Second, the algorithm was operated at 0 °C, 10 °C, and 40 °C to assess its temperature adaptability.

The experimental data originated from the publicly available dataset of the CALCE Battery Group at the University of Maryland [37], which was obtained by simulating real-world electric vehicle driving scenarios. This dataset includes dynamic battery operational data (voltage, current, and temperature) under three typical driving profiles—DST, FUDS, and US06—tested at different environmental temperatures.

This paper employs the maximum error (MaxError), the root mean squared error (RMSE), and the mean absolute error (MAE) as evaluation metrics, which are expressed, respectively, as follows:

$$\begin{cases} \text{MaxError} = \max_{i=1,\ldots,n} |y_i - \hat{y}_i| \\ \text{RMSE} = \sqrt{\dfrac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \\ \text{MAE} = \dfrac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \end{cases} \tag{37}$$

where y_i and $\hat{y}_i$ denote the true value and the predicted value, respectively, and n is the number of samples.

7.2. OCV-SOC Curve Fitting and PSO-Based Parameter Identification

During low-rate charge–discharge testing, the battery's polarization effects (ohmic resistance, concentration polarization) are minimal, and the measured voltage approaches a quasi-equilibrium state. In this experiment, charge–discharge testing was conducted at a constant current rate of $C/20$ (where C denotes the battery's nominal capacity). This rate effectively suppresses polarization effects, thereby providing a reliable basis for establishing an accurate OCV-SOC relationship. Voltage data acquired at various temperatures are well-suited as a reference for establishing the OCV-SOC relationship. Based on discharge

voltage curves at each temperature, polynomial fitting was applied to derive OCV-SOC curves. Fitting accuracy depends on polynomial order: while higher orders generally improve precision, excessively high orders may cause overfitting, which is manifested as terminal "oscillations" that compromise practical applicability. Conversely, insufficient orders result in underfitting, leading to unacceptable curve deviations. As shown in the fitting comparison and error evaluation results of polynomial functions of different orders in Figures 8 and 9, the fitting accuracy generally demonstrates an upward trend as the order increases. The ninth-order polynomial significantly outperforms lower-order models, while its error level closely matches that of the tenth-order model. By ensuring precision, the ninth-order polynomial exhibits superior model stability and effectively mitigates the risk of overfitting, and has therefore been selected as the final choice.

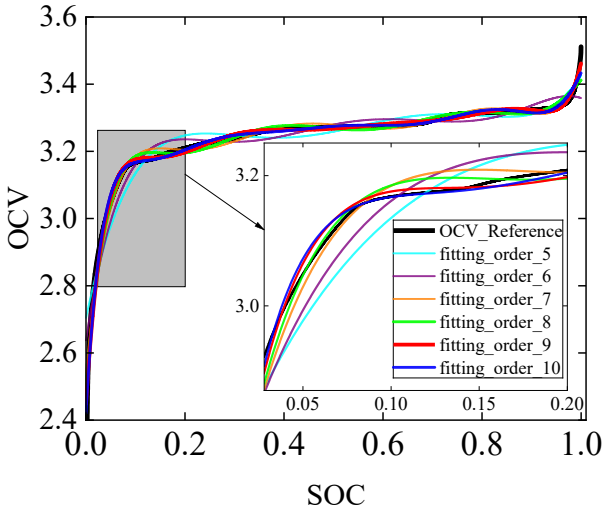


Figure 8. OCV-SOC fitting curve comparison chart.

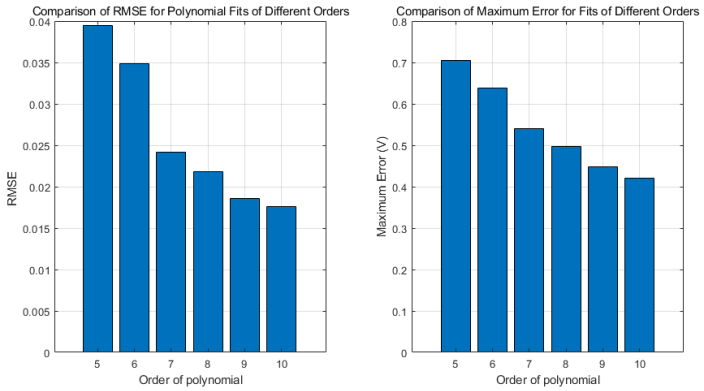


Figure 9. OCV-SOC fitting curve error analysis chart.

The polynomial equation is expressed as

$$U_{OCV}(\mathrm{SOC}) = P_1 \cdot \mathrm{SOC}^9 + \ldots + P_9 \cdot \mathrm{SOC}^1 + P_{10} \tag{38}$$

The parameter identification for the equivalent circuit model (ECM) based on the particle swarm optimization (PSO) algorithm was validated under the FUDS driving cycle at an ambient temperature of 20 °C. The identification process utilized a dedicated parameter identification dataset (including dynamic profiles such as DST, FUDS, and US06), which is independent of the data used for model training, with no overlap in time or cycle number with the subsequent BiLSTM training datasets. The PSO algorithm was configured as follows: population size of 100 and a maximum of 200 iterations to ensure

convergence and reproducibility of the results. The model parameters obtained through this algorithm are listed in Table 1. The terminal voltage values calculated using these parameters were compared against actual measurements, as shown in Figure 10, which also includes the voltage error curve. The results demonstrate that the calculated voltage curve closely matches the measured values, confirming the validity and accuracy of the identified parameters.

Table 1. Model parameter identification results.

Temperature	R_0 (Ω)	R_1 (Ω)	C_1 (F)	R_2 (Ω)	C_2 (F)
20 °C	0.17206	0.014629	651.4273	0.0012667	9683.0963

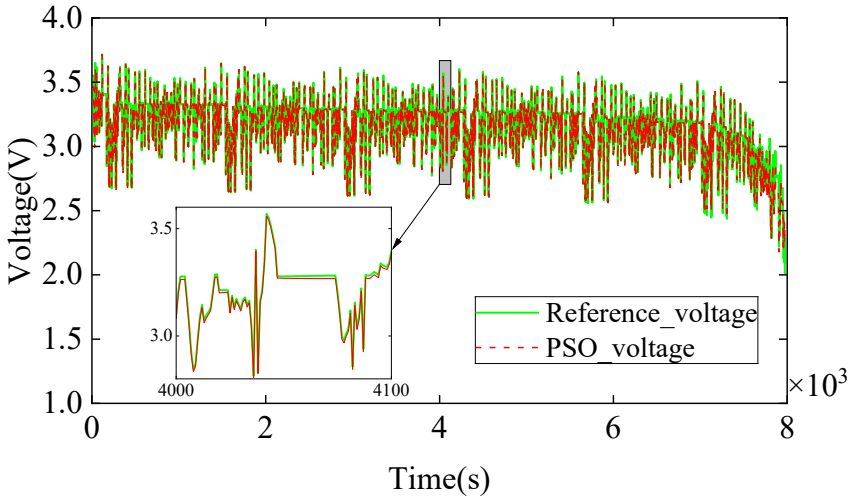


Figure 10. Comparative analysis of terminal voltage under FUDS driving.

7.3. Accuracy Validation of SOC Estimation Based on BiLSTM-AUKF

For the BO-BiLSTM-AUKF method proposed in this paper, we employ Bayesian optimization to systematically search five key hyperparameters over 100 iterations, covering both network architecture and regularization configurations. The architectural hyperparameters include the number of units in the LSTM layer (64–512), the number of units in the first fully connected layer (128–512), and the number of units in the second fully connected layer (32–128). The regularization hyperparameters consist of the dropout rate (0.1–0.5) and the L_2 regularization weight coefficient (searched on a logarithmic scale from 10^{-6} to 10^{-2}). This parameter space strikes a balance between model expressive power and regularization requirements, facilitating a systematic exploration of network structures with varying complexities and their corresponding regularization settings. To ensure fair comparisons, all data-driven models (including LSTM and BiLSTM) adopted a unified architectural design. During training, the Adam optimizer was used with an initial learning rate of 0.001 and a periodic decay strategy, while the batch size was fixed at 512. Training halted early if the validation loss showed no improvement over 20 consecutive epochs. The accuracy and generalization capability of the trained models were ultimately evaluated on the testing set.

To verify the estimation accuracy of BiLSTM and the performance improvement effect of the Sage–Husa adaptive strategy on UKF, this study designed five comparative experiments: under a constant temperature of 20 °C, using the same DST, FUDS, and US06 driving cycle data, the GRU, LSTM, BiLSTM, BiLSTM-UKF, and BiLSTM-AUKF algorithms were, respectively, employed for SOC estimation. Notably, to ensure a rigorous evaluation of generalization performance under the cross-profile validation framework,

each model was trained on a combination of the other two driving profiles, distinct from the testing profile. Specifically, models tested on DST (Figure 11 and Table 2) were trained on US06 and FUDS data; models tested on FUDS (Figure 12 and Table 3) were trained on DST and US06 data; and models tested on US06 (Figure 13 and Table 4) were trained on DST and FUDS data. In addition, all models were trained using multi-temperature operating condition data ranging from −10 °C to 50 °C. A 30-timestep sliding window and min-max normalization were applied for time-series reconstruction and preprocessing to enhance the models' environmental adaptability and training efficiency. Figures 11–13 present the estimation curves and error distributions of the five algorithms under the three driving cycles, while Tables 2–4 provide a quantitative comparison using three key metrics (MaxError, MAE, and RMSE), clearly quantifying the synergistic gains from bidirectional structure optimization and adaptive noise correction.

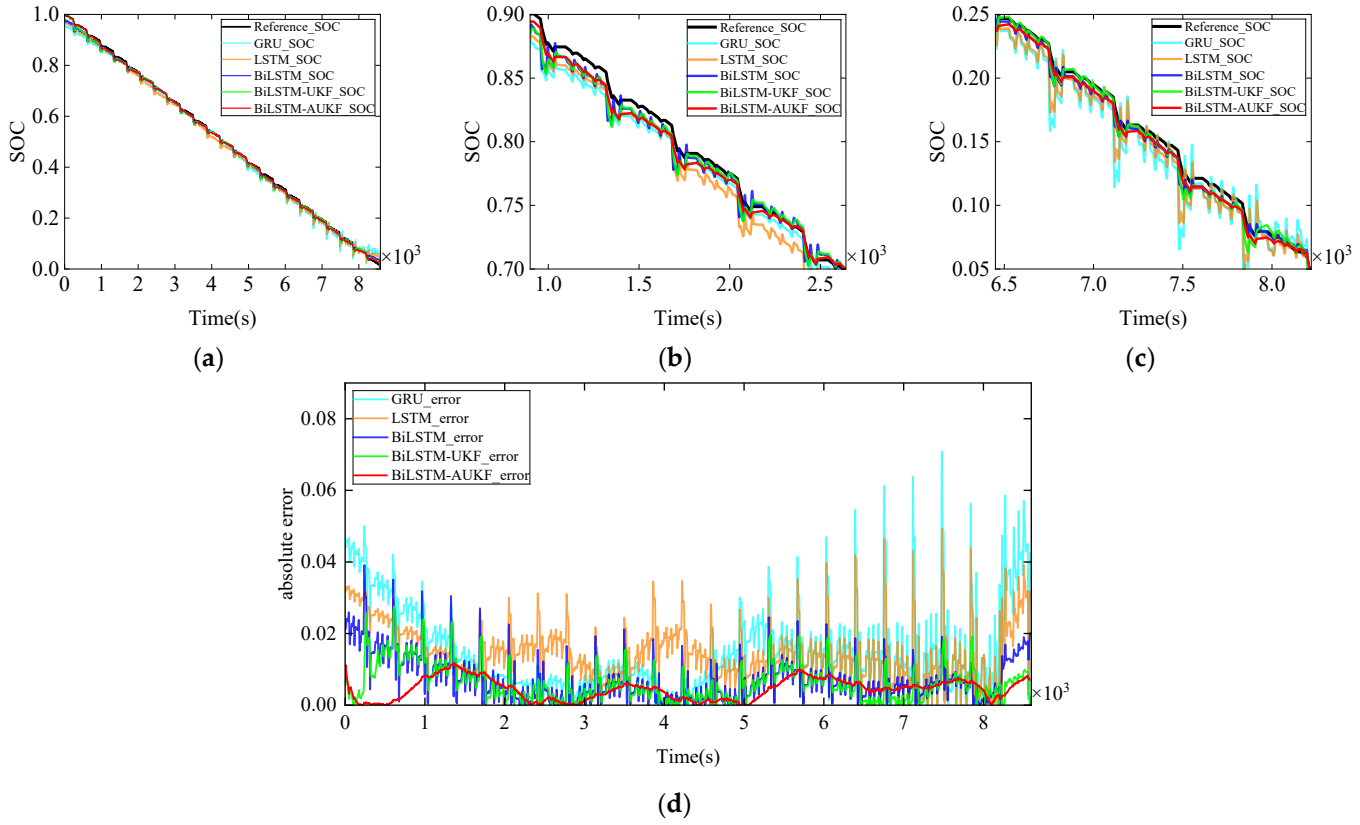


Figure 11. SOC estimation results and errors with models trained on the combined US06 and FUDS datasets and tested on the DST profile: (**a**) SOC estimation for the entire DST working condition process; (**b**) partial enlarged view of the front section of (**a**); (**c**) partial enlarged view of the rear section of (**a**); (**d**) absolute error comparison across algorithms.

Table 2. SOC estimation error statistics under the DST test profile for models trained on the combined US06 and FUDS driving cycles.

Working Conditions	Estimation Method	MaxError/%	MAE/%	RMSE/%
DST	GRU	7.0895	1.5494	1.9514
	LSTM	4.9188	1.4984	1.6779
	BiLSTM	3.9136	0.73297	0.94334
	BiLSTM-UKF	2.7522	0.60915	0.78313
	BiLSTM-AUKF	1.1446	0.44287	0.53123

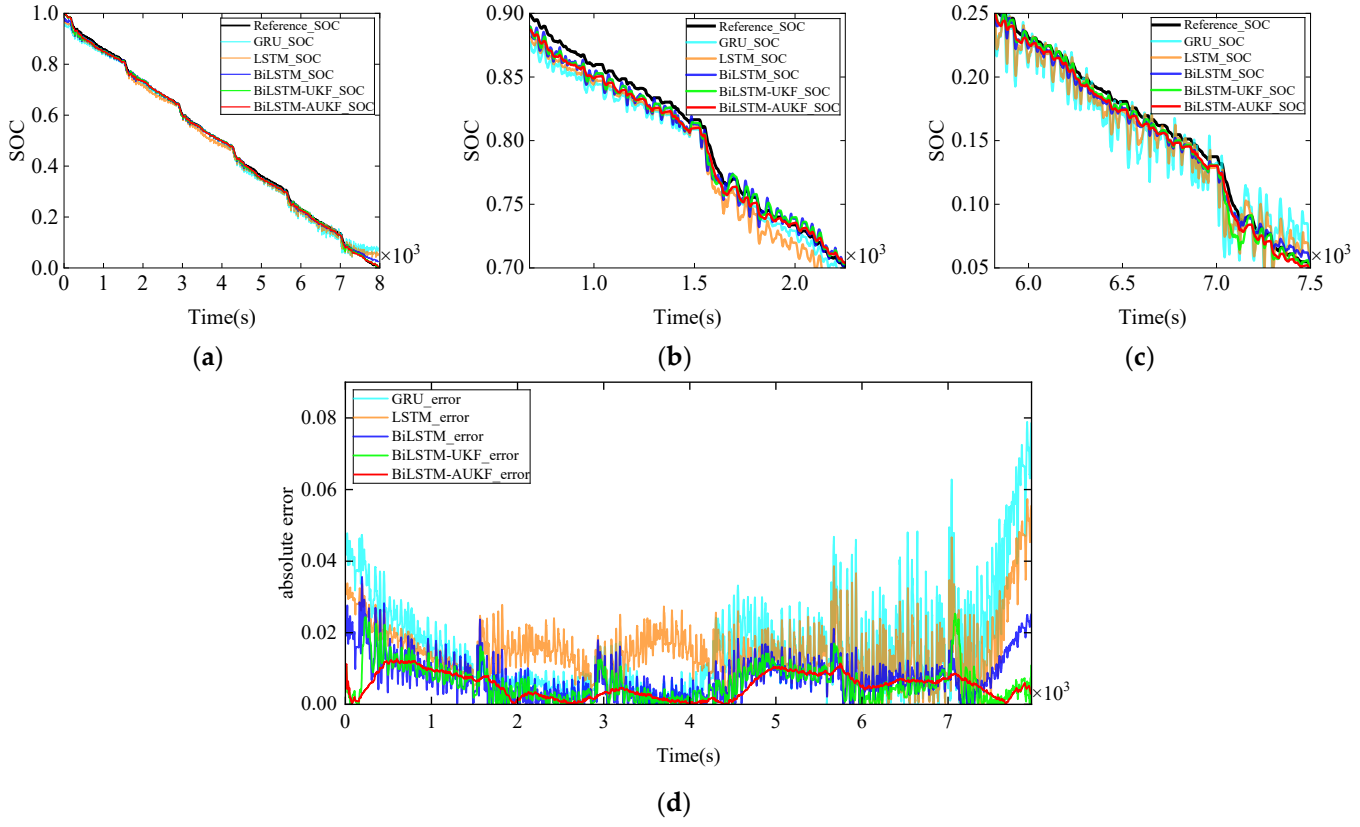


Figure 12. SOC estimation results and errors with models trained on the combined US06 and DST datasets and tested on the FUDS profile: (**a**) SOC estimation for the entire FUDS working condition process; (**b**) partial enlarged view of the front section of (**a**); (**c**) partial enlarged view of the rear section of (**a**); (**d**) absolute error comparison across algorithms.

Table 3. SOC estimation error statistics under the FUDS test profile for models trained on the combined US06 and DST driving cycles.

Working Conditions	Estimation Method	MaxError/%	MAE/%	RMSE/%
FUDS	GRU	7.8898	1.601	2.1161
	LSTM	5.7338	1.5431	1.7605
	BiLSTM	3.557	0.74094	0.93266
	BiLSTM-UKF	2.532	0.58893	0.75801
	BiLSTM-AUKF	1.265	0.52662	0.62845

Comprehensive experimental evaluations confirm the superior estimation accuracy of the proposed BiLSTM-AUKF algorithm across diverse operating conditions compared to conventional methods such as GRU and LSTM. Under the dynamic stress test (DST) profile, for instance, the baseline GRU architecture yields an RMSE of 1.9514%, whereas the LSTM reduces this value to 1.6779%. The bidirectional LSTM structure demonstrates further improvement, achieving RMSE and MAE values of 0.94334% and 0.73297%, respectively, underscoring its enhanced capability in capturing temporal dependencies compared to unidirectional counterparts. Further refinement through the adaptive unscented Kalman filter culminates in the BiLSTM-AUKF hybrid model attaining RMSE and MAE values of 0.53123% and 0.44287%, corresponding to accuracy improvements of approximately 32.16% and 27.29% relative to the BiLSTM-UKF framework. These results substantiate that the synergistic integration of BiLSTM's sophisticated feature extraction with AUKF's robust filtering capacity produces a substantial enhancement in SOC estimation precision.

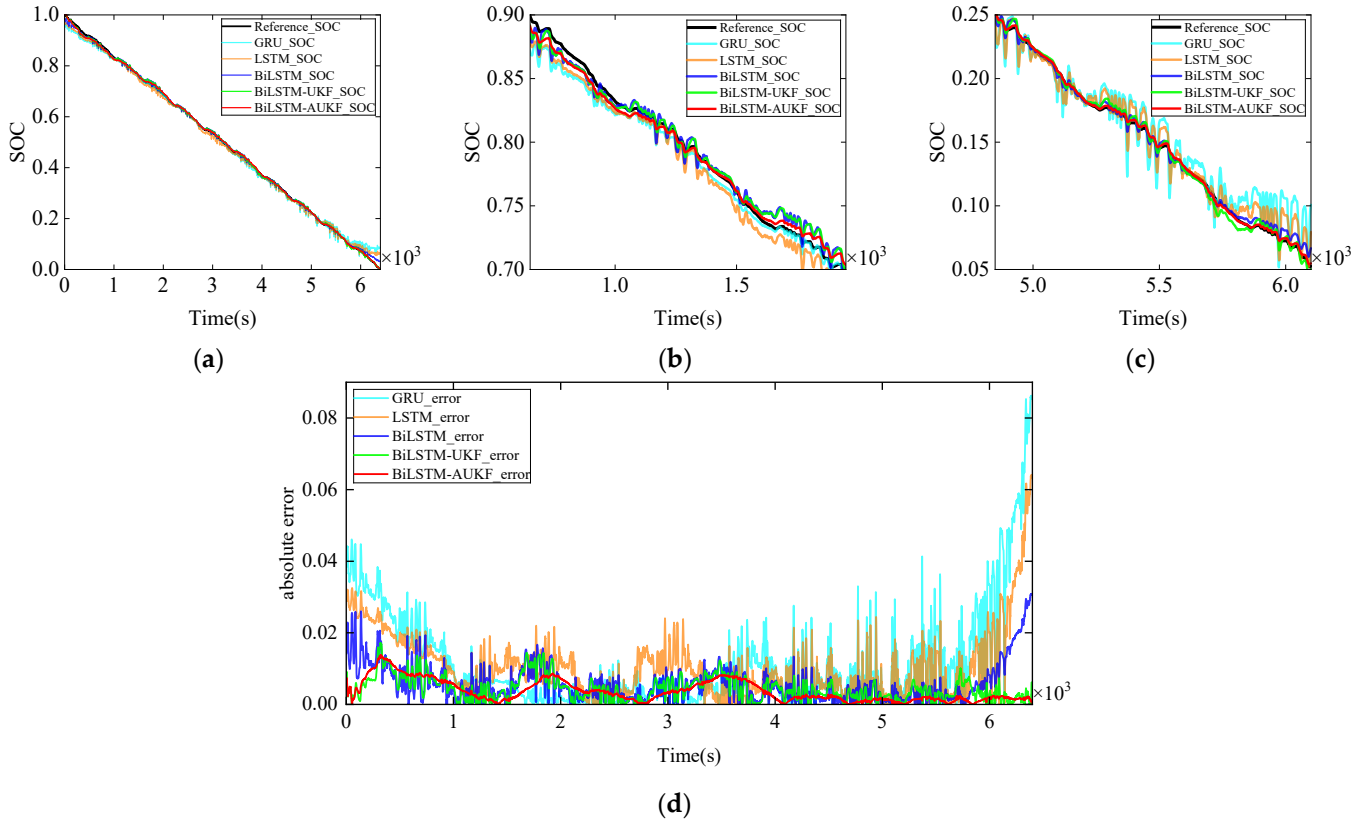


Figure 13. SOC estimation results and errors with models trained on the combined DST and FUDS datasets and tested on the US06 profile: (**a**) SOC estimation for the entire US06 working condition process; (**b**) partial enlarged view of the front section of (**a**); (**c**) partial enlarged view of the rear section of (**a**); (**d**) absolute error comparison across algorithms.

Table 4. SOC estimation error statistics under the US06 test profile for models trained on the combined DST and FUDS driving cycles.

Working Conditions	Estimation Method	MaxError/%	MAE/%	RMSE/%
US06	GRU	8.6252	1.2165	1.8535
	LSTM	6.4148	1.0723	1.4226
	BiLSTM	3.0888	0.58034	0.78363
	BiLSTM-UKF	1.7188	0.43867	0.54656
	BiLSTM-AUKF	1.3768	0.37033	0.47677

While estimation accuracy serves as the primary optimization objective, computational efficiency remains a crucial practical concern. As summarized in Table 5, a systematic evaluation was performed to assess the training and inference costs associated with different algorithmic architectures.

As shown in Table 5, a quantitative comparison of the computational efficiency of the algorithms was conducted under a strict cross-condition validation framework. Our analysis confirms that the superior estimation accuracy of the BiLSTM-AUKF model comes with a manageable increase in computational load. It is true that BiLSTM-based architectures demand more processing time than simpler GRU or LSTM networks. However, the critical refinement introduced by the AUKF contributes negligibly to the overall latency. This produces a highly favorable trade-off, as the algorithm achieves a decisive leap in estimation quality with significantly lower RMSE and MAE, at only a modest and manageable cost. In safety-critical BMS applications where precision is paramount, such an investment in computational resources is unequivocally justified.

Table 5. Efficiency comparison table.

Algorithm	Training Set	Epoch Time (s)	Testing Set	Inference Time per Time Step (s)
GRU	DST + FUDS	67.33	US06	0.0087387
	DST + US06	70.10	FUDS	0.008614
	US06 + FUDS	68.67	DST	0.0085396
LSTM	DST + FUDS	71.35	US06	0.0090138
	DST + US06	73.40	FUDS	0.0089999
	US06 + FUDS	73.02	DST	0.008971
BiLSTM	DST + FUDS	74.45	US06	0.013268
	DST + US06	76.67	FUDS	0.013047
	US06 + FUDS	76.49	DST	0.013134
BiLSTM-UKF	DST + FUDS	74.45	US06	0.013296473
	DST + US06	76.67	FUDS	0.013074086
	US06 + FUDS	76.49	DST	0.013159713
BiLSTM-AUKF	DST + FUDS	74.45	US06	0.013297232
	DST + US06	76.67	FUDS	0.013075712
	US06 + FUDS	76.49	DST	0.013164777

7.4. Generalization Capability Validation for BiLSTM-AUKF-Based SOC Estimation

To further validate the robustness of the proposed BiLSTM-AUKF algorithm under inaccurate initial SOC conditions, experiments were conducted at 20 °C using the DST, FUDS, and US06 driving cycles with varied initial SOC settings. Notably, the experimental conditions remained consistent with those described in Section 7.3, with the only modification being the variation in initial SOC values. The true initial SOC was 100%, while the algorithm was initialized at 90%, 80%, and 70%, respectively. As shown in Figure 14, the BiLSTM-AUKF algorithm rapidly converges to the true SOC value despite initial inaccuracies, demonstrating exceptional robustness.

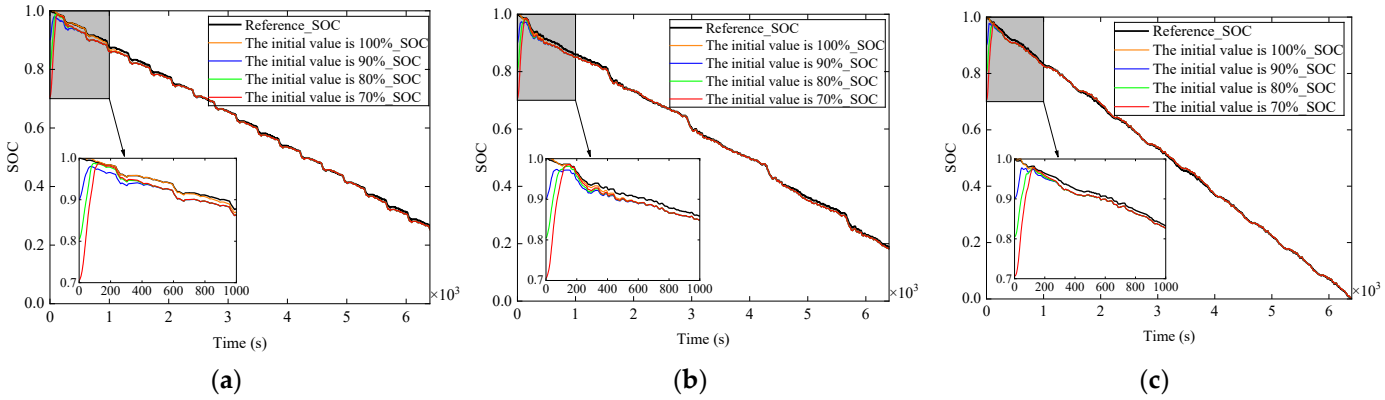


Figure 14. SOC estimation with varying Initial values under driving cycles of (**a**) DST, (**b**) FUDS, and (**c**) US06.

To systematically validate the temperature adaptability of the BiLSTM-AUKF algorithm across a wide thermal range, this study selected three characteristic temperature points: 0 °C, 10 °C, and 40 °C. Using full drive-cycle data (DST, FUDS, and US06) from the CALCE dataset, temperature robustness tests were conducted. It should be noted that the model training protocols remained identical to those described in Section 7.3, with the only extension being the incorporation of additional temperature test points. As detailed in Tables 6–8, three core metrics—MaxError, MAE, and RMSE—were statistically analyzed to quantify the algorithm's precision degradation characteristics and stability boundaries under thermal stress, conclusively demonstrating its wide-temperature adaptation mechanism. Validation confirms that the proposed integrated algorithm achieves high-precision

SOC estimation while maintaining robust performance across diverse environmental temperatures and operational conditions.

Table 6. SOC estimation errors for different driving cycles at 0 °C.

Working Condition	Estimation Method	MaxError/%	MAE/%	RMSE/%
DST	LSTM	4.6087	1.467	1.6347
	BiLSTM	3.3173	0.71732	0.89197
	BiLSTM-UKF	2.6382	0.70204	0.86481
	BiLSTM-AUKF	1.6001	0.51654	0.61679
FUDS	LSTM	6.2205	1.4044	1.6845
	BiLSTM	3.315	0.64464	0.85691
	LSTM-UKF	2.9059	0.61586	0.80715
	BiLSTM-AUKF	2.1648	0.4234	0.54408
US06	LSTM	4.8938	1.0087	1.3199
	BiLSTM	2.3827	0.52417	0.69309
	LSTM-UKF	2.6392	0.48736	0.6562
	BiLSTM-AUKF	1.6945	0.31124	0.39664

Table 7. SOC estimation errors for different driving cycles at 10 °C.

Working Condition	Estimation Method	MaxError/%	MAE/%	RMSE/%
DST	LSTM	4.9595	1.6323	1.8089
	BiLSTM	3.4098	0.79409	0.97419
	BiLSTM-UKF	2.6918	0.77723	0.94338
	BiLSTM-AUKF	1.9329	0.60588	0.72084
FUDS	LSTM	4.4645	1.4965	1.6572
	BiLSTM	3.4287	0.70729	0.88551
	LSTM-UKF	2.7774	0.69009	0.8514
	BiLSTM-AUKF	1.4638	0.57714	0.68095
US06	LSTM	6.1901	1.1609	1.5187
	BiLSTM	2.9115	0.55783	0.76841
	LSTM-UKF	1.768	0.41938	0.52105
	BiLSTM-AUKF	1.3377	0.33496	0.43993

Table 8. SOC estimation errors for different driving cycles at 40 °C.

Working Condition	Estimation Method	MaxError/%	MAE/%	RMSE/%
DST	LSTM	5.7929	1.4962	1.7191
	BiLSTM	3.5519	0.71453	0.92059
	BiLSTM-UKF	2.4396	0.54714	0.7102
	BiLSTM-AUKF	0.94416	0.40777	0.47844
FUDS	LSTM	6.0939	1.5251	1.7302
	BiLSTM	3.5385	0.71447	0.91028
	LSTM-UKF	2.5037	0.57598	0.71939
	BiLSTM-AUKF	1.1728	0.47813	0.56799
US06	LSTM	6.0055	1.1943	1.5234
	BiLSTM	2.7287	0.57078	0.76258
	LSTM-UKF	1.6306	0.46982	0.56751
	BiLSTM-AUKF	1.1043	0.35783	0.45392

8. Conclusions

This paper proposes a high-precision SOC estimation method for lithium-ion batteries based on a BiLSTM-AUKF synergistic architecture. Experimental validation demonstrates that under diverse temperature environments and complex operating conditions, the BiLSTM-AUKF algorithm notably outperforms traditional Kalman filtering and neural network approaches in SOC estimation accuracy. Particularly under significant initial SOC errors, the algorithm exhibits exceptional convergence speed and robustness, rapidly approaching the true values. The BiLSTM's powerful time-series feature extraction capability effectively compensates for model mismatch, while the Sage–Husa adaptive strategy dynamically adjusts the AUKF's noise statistics, substantially reducing dependency on model and observation noise sensitivity. Consequently, this approach overcomes the limitations of conventional methods—parameter mismatch, error accumulation, and noise-induced accuracy degradation. The proposed method provides a novel framework for high-precision BMS, with its enhanced SOC estimation accuracy offering significant engineering value in optimizing energy management strategies and extending battery cycle life.

However, practical implementation in BMS faces challenges including high computational demands, generalization across diverse battery chemistries and aging states, and dynamic noise management. Future research should focus on developing lightweight algorithm versions, incorporating transfer learning for better adaptability, and enabling joint SOC-SOH estimation. Further validation under real electric vehicle operating conditions is equally essential for practical deployment.

Author Contributions: Conceptualization, R.W. and L.L.; methodology, R.W.; software, R.W.; validation, L.L., R.W., and F.Y.; formal analysis, H.Z., Q.Q. (Qifeng Qian), Q.Q. (Qiansheng Qiu) and L.X.; investigation, R.W., L.L., and C.T.; resources, R.W.; data curation, L.L. and R.W.; writing—original draft preparation, R.W. and L.L.; writing—review and editing, R.W. and F.Y.; visualization, L.L.; supervision, R.W. and C.T.; project administration, R.W.; funding acquisition, R.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (grant number 62301086), the Natural Science Foundation of Jiangsu Province (grant number BK20230627), and the High-Level Talent Introduction Project of Changzhou University, China (grant number ZMF22020078).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Rui Wang, Honghou Zhang, Qifeng Qian, Lingchao Xiao, Qiansheng Qiu were employed by the company Zhejiang Sunoren Solar Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hannan, M.A.; Lipu, M.S.H.; Hussain, A.; Mohamed, A. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renew. Sustain. Energy Rev.* **2017**, *78*, 834–854. [CrossRef]
2. Wang, Y.; Zhang, X.; Li, K.; Zhao, G.; Chen, Z. Perspectives and challenges for future lithium-ion battery control and management. *eTransportation* **2023**, *18*, 100260. [CrossRef]
3. Zhao, X.; Qian, X.; Xuan, D.; Jung, S. State of charge estimation of lithium-ion battery based on multi-input extreme learning machine using online model parameter identification. *J. Energy Storage* **2022**, *56*, 105796. [CrossRef]
4. He, H.; Zhang, X.; Xiong, R.; Xu, Y.; Guo, H. Online model-based estimation of state-of-charge and open-circuit voltage of lithium-ion batteries in electric vehicles. *Energy* **2012**, *39*, 310–318. [CrossRef]
5. Pillai, P.; Sundaresan, S.; Kumar, P.; Pattipati, K.R.; Balasingam, B. Open-Circuit Voltage Models for Battery Management Systems: A Review. *Energies* **2022**, *15*, 6803. [CrossRef]

6. Zhang, X.; Duan, L.; Gong, Q.; Wang, Y.; Song, H. State of charge estimation for lithium-ion battery based on adaptive extended Kalman filter with improved residual covariance matrix estimator. *J. Power Sources* **2024**, *589*, 233758. [CrossRef]
7. Shrivastava, P.; Soon, T.K.; Bin Idris, M.Y.I.; Mekhilef, S.; Adnan, S.B.R.S. Comprehensive co-estimation of lithium-ion battery state of charge, state of energy, state of power, maximum available capacity, and maximum available energy. *J. Energy Storage* **2022**, *56*, 106049. [CrossRef]
8. Luan, Z.; Qin, Y.; Hu, B.; Zhao, W.; Wang, C. Estimation of state of charge for hybrid unmanned aerial vehicle Li-ion power battery for considering rapid temperature change. *J. Energy Storage* **2023**, *59*, 106479. [CrossRef]
9. Wang, L.; Han, J.; Liu, C.; Li, G. State of charge estimation of lithium-ion based on VFFRLS-noise adaptive CKF algorithm. *Ind. Eng. Chem. Res.* **2022**, *61*, 22. [CrossRef]
10. Li, S.; Li, Y.; Zhao, D.; Zhang, C. Adaptive state of charge estimation for lithium-ion batteries based on implementable fractional-order technology. *J. Energy Storage* **2020**, *32*, 101838. [CrossRef]
11. Wu, C.; Hu, W.; Meng, J.; Xu, X.; Huang, X.; Cai, L. State-of-charge estimation of lithium-ion batteries based on MCC-AEKF in non-Gaussian noise environment. *Energy* **2023**, *274*, 127316. [CrossRef]
12. Zhao, H.; Liao, C.; Zhang, C.; Wang, L.; Wang, L. State-of-charge estimation of lithium-ion battery: Joint long short-term memory network and adaptive extended Kalman filter online estimation algorithm. *J. Power Sources* **2024**, *604*, 234451. [CrossRef]
13. Wu, S.; Pan, W.; Zhu, M. A collaborative estimation scheme for lithium-ion battery state of charge and state of health based on electrochemical model. *J. Electrochem. Soc.* **2022**, *169*, 090516. [CrossRef]
14. Shen, D.; Ding, J.; Hao, T. Elman neural network and thevenin equivalent circuit model based multi-measurement Kalman filter for SOC estimation. *Ionics* **2024**, *30*, 833–845. [CrossRef]
15. Dar, T.H.; Singh, S. Advanced integration of bidirectional long short-term memory neural networks and innovative extended Kalman filter for state of charge estimation of lithium-ion battery. *J. Power Sources* **2025**, *628*, 235893. [CrossRef]
16. Monirul, I.M.; Qiu, L.; Ruby, R. Accurate SOC estimation of ternary lithium-ion batteries by HPPC test-based extended Kalman filter. *J. Energy Storage* **2024**, *92*, 112304. [CrossRef]
17. Wang, M.; Wang, G.; Xiao, Z.; Sun, Y.; Zheng, Y. State of charge estimation of LiFePO4 in various temperature scenarios. *Batteries* **2023**, *9*, 43. [CrossRef]
18. Li, H.; Fu, L.; Long, X.; Liu, L.; Zeng, Z. A hybrid deep learning model for lithium-ion batteries state of charge estimation based on quantile regression and attention. *Energy* **2024**, *294*, 130834. [CrossRef]
19. Zhang, C.; Wang, S.; Yu, C.; Xie, Y.; Fernandez, C. Novel improved particle swarm optimization- extreme learning machine algorithm for state of charge estimation of lithium-ion batteries. *Ind. Eng. Chem. Res.* **2022**, *6*, 46. [CrossRef]
20. Ma, L.; Zhang, T. Deep learning-based battery state of charge estimation: Enhancing estimation performance with unlabelled training samples. *J. Energy Chem.* **2023**, *80*, 48–57. [CrossRef]
21. Che, Y.; Xu, L.; Teodorescu, R.; Hu, X.; Onori, S. Enhanced SOC estimation for LFP batteries: A synergistic approach using coulomb counting reset, machine learning, and relaxation. *ACS Energy Lett.* **2025**, *63*, 2. [CrossRef]
22. Hao, X.; Wang, S.; Fan, Y.; Xie, Y.; Fernandez, C. An improved forgetting factor recursive least square and unscented particle filtering algorithm for accurate lithium-ion battery state of charge estimation. *J. Energy Storage* **2023**, *59*, 106478. [CrossRef]
23. Wang, C.; Li, Q.; Tang, A.; Zhang, Z. A comparative study of state of charge estimation methods of ultracapacitors for electric vehicles considering temperature characteristics. *J. Energy Storage* **2023**, *63*, 106908. [CrossRef]
24. El Fallah, S.; Kharbach, J.; Hammouch, Z.; Rezzouk, A.; Jamil, M.O. State of charge estimation of an electric vehicle's battery using deep neural Networks: Simulation and experimental results. *J. Energy Storage* **2023**, *62*, 106904. [CrossRef]
25. Liu, X.; Li, Q.; Wang, L.; Lin, M.; Wu, J. Data-driven state of charge estimation for power battery with improved extended Kalman filter. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1500910. [CrossRef]
26. Wadi, A.; Abdel-Hafez, M.F.; Hussein, A.; Alkhawaja, F. Alleviating dynamic model uncertainty effects for improved battery SOC estimation of EVs in highly dynamic environments. *IEEE Trans. Veh. Technol.* **2021**, *70*, 6554–6566. [CrossRef]
27. Zhang, X.; Hou, J.; Wang, Z.; Jiang, Y. Study of SOC estimation by the ampere-hour integral method with capacity correction based on LSTM. *Batteries* **2022**, *8*, 170. [CrossRef]
28. Yang, F.; Zhang, S.; Li, W.; Miao, Q. State-of-charge estimation of lithium-ion batteries using LSTM and UKF. *Energy* **2020**, *201*, 117664. [CrossRef]
29. Xu, K.; He, T.; Yang, P.; Meng, X.; Zhu, C.; Jin, X. A new online SOC estimation method using broad learning system and adaptive unscented Kalman filter algorithm. *Energy* **2024**, *309*, 132920. [CrossRef]
30. Tian, J.; Xiong, R.; Lu, J.; Chen, C.; Shen, W. Battery state-of-charge estimation amid dynamic usage with physics-informed deep learning. *Energy Storage Mater.* **2022**, *50*, 718–729. [CrossRef]
31. Takyi-Aninakwa, P.; Wang, S.; Liu, G.; Bage, A.N.; Masahudu, F.; Guerrero, J.M. An enhanced lithium-ion battery state-of-charge estimation method using long short-term memory with an adaptive state update filter incorporating battery parameters. *Eng. Appl. Artif. Intell.* **2024**, *132*, 107946. [CrossRef]

32. Chen, Z.; Zhao, H.; Shu, X.; Zhang, Y.; Shen, J.; Liu, Y. Synthetic state of charge estimation for lithium-ion batteries based on long short-term memory network modeling and adaptive H-Infinity filter. *Energy* **2021**, *228*, 120630. [CrossRef]
33. Wang, Q.; Ye, M.; Wei, M.; Lian, G.; Li, Y. Deep convolutional neural network based closed-loop SOC estimation for lithium-ion batteries in hierarchical scenarios. *Energy* **2023**, *263*, 125718. [CrossRef]
34. Chen, J.; Zhang, Y.; Li, W.; Cheng, W.; Zhu, Q. State of charge estimation for lithium-ion batteries using gated recurrent unit recurrent neural net-work and adaptive Kalman filter. *J. Energy Storage* **2022**, *55*, 105396. [CrossRef]
35. Wei, M.; Ye, M.; Zhang, C.; Lian, G.; Xia, B.; Wang, Q. Robust state of charge estimation of LiFePO4 batteries based on Sage-Husa adaptive Kalman filter and dynamic neural network. *Electrochim. Acta* **2024**, *477*, 143778. [CrossRef]
36. Yang, Y.; Xu, Y.; Nie, Y.; Li, J.; Liu, S.; Zhao, L.; Yu, Q.; Zhang, C. Deep transfer learning enables battery state of charge and state of health estimation. *Energy* **2024**, *294*, 130779. [CrossRef]
37. Tian, Y.; Lai, R.; Li, X.; Xiang, L.; Tian, J. 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* **2020**, *265*, 114789. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.