

Article

End-Cloud Collaboration Approach for State-of-Charge Estimation in Lithium Batteries Using CNN-LSTM and UKF

Wentao Wang¹, Bin Ma^{1,*}, Xiao Hua², Bosong Zou³, Lisheng Zhang¹, Hanqing Yu¹, Kaiyi Yang¹, Shichun Yang¹ and Xinhua Liu^{1,4}

¹ School of Transportation Science and Engineering, Beihang University, Beijing 102206, China

² State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing 100084, China

³ China Software Testing Center, Beijing 100038, China

⁴ Dyson School of Design Engineering, Imperial College London, Exhibition Road, South Kensington Campus, London SW7 2AZ, UK

* Correspondence: mabin_wayne@sina.cn

Abstract: The accurate estimation of the state of charge (SOC) plays a crucial role in ensuring the range of electric vehicles (EVs) and the reliability of the EVs battery. However, due to the dynamic working conditions in the implementation of EVs and the limitation of the onboard BMS computational force, it is challenging to achieve a reliable, high-accuracy and real-time online battery SOC estimation under diverse working scenarios. Therefore, this study proposes an end-cloud collaboration approach of lithium-ion batteries online estimate SOC. On the cloud-side, a deep learning model constructed based on CNN-LSTM is deployed, and on the end-side, the coulomb counting method and Kalman's filter are deployed. The estimation results at both sides are fused through the Kalman filtering algorithm, realizing high-accuracy and real-time online estimation of SOC. The proposed approach is evaluated with three dynamic driving profiles and the results demonstrate the proposed approach has high accuracy under different temperatures and initial errors, where the root means square error (RMSE) is lower than 1.5% and the maximum error is lower than 5%. Furthermore, this method could achieve high-accuracy and real-time SOC online estimation under the cyber hierarchy and interactional network (CHAIN) framework and can be extended to multi-state collaborative online estimation.

Keywords: state of charge; convolutional neural network; long short-term memory; unscented Kalman filter; end-cloud collaboration



Citation: Wang, W.; Ma, B.; Hua, X.; Zou, B.; Zhang, L.; Yu, H.; Yang, K.; Yang, S.; Liu, X. End-Cloud Collaboration Approach for State-of-Charge Estimation in Lithium Batteries Using CNN-LSTM and UKF. *Batteries* **2023**, *9*, 114. <https://doi.org/10.3390/batteries9020114>

Academic Editor: Catia Arbizzani

Received: 24 December 2022

Revised: 26 January 2023

Accepted: 3 February 2023

Published: 6 February 2023



Copyright: © 2023 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

With growing concern over the environment and the greenhouse effect, EVs are gradually becoming the focus of attention. Lithium-ion batteries are widely applied for the power system of EVs due to their excellent performance, i.e., high energy density, high voltage and long life [1–4]. Lithium-ion batteries are easily influenced by ambient temperature, charge and discharge rate, aging and other factors under operations [5–7]. It is essential to establish a reliable battery management system (BMS) to monitor and predict the battery state with high-accuracy and real-time [8,9]. Among the states of batteries, the battery state of charge represents the ratio of the remaining usable capacity of the battery to the applicable current maximum capacity, which is one of the key states of the battery. Accurate and real-time SOC information determines the range and ensure the reliability of the battery operation by BMS [10]. However, the battery is a highly complex electrochemical system with time-varying and non-linear characteristics because of the complex electrochemical dynamics and multi-physical field coupling inside the battery [11–13]. Therefore, it is still a significant task to accurately estimate SOC online based on macroscopic signals.

At present, the essential SOC estimation methods include: 1. the coulomb counting method (CCM), 2. the open-circuit-voltage (OCV) method, 3. the model-based method and 4. the data-driven method.

The coulomb counting method integrates the measured current with time, while monitoring the real-time SOC of the battery online based on the initial SOC. This method is equipped with the ease of implementation, but the initial error will affect the accuracy seriously. Additionally, such method is an open-loop method, where the cumulative error influences the SOC error [14].

By establishing the corresponding relationship between the OCV and SOC, the OCV method can map the SOC of the battery by obtaining the OCV [15,16]. However, the OCV-SOC relationship will be influenced by temperature and degradation. If the relationship is not updated in time, serious SOC estimation errors may result. Moreover, when the SOC is in the middle range (20–80%), the OCV-SOC characteristic curve will present a flat plateau, especially in lithium iron phosphate (LFP) batteries. At this time, the small error of OCV will sharply increase the estimation error of SOC, which will cause the estimation to be very sensitive. In addition, when measuring OCV, it is necessary to disconnect the battery for a certain period of time (more than 2 h) to make it reach a stable state to ensure accuracy. Therefore, this method is only applicable to specific application scenario such as laboratories, and cannot achieve the requirements of onboard applications [17].

The model-based method is mainly implemented through three steps: battery modelling, parameter identification and state estimation. At present, most of the model-based methods used in practice are to establish an equivalent circuit model (ECM) and identify the parameters, and then estimate the SOC through Kalman's filter or other observers [18]. The model-based method belongs to a closed-loop system, which can reduce the interference of uncertainty through self-tuning. It has robustness to initial SOC offset and measurement noise and has achieved high estimation accuracy under specific operating conditions. Plett applied extended Kalman filters (EKF) and different battery models in BMS to estimate the SOC for lithium polymer batteries [19–21]. Based on the cubature Kalman filter (CKF), Peng et al. [22] achieved SOC estimation with high-accuracy and robustness by establishing an equivalent circuit model and identifying the parameters using the Sequential Quadratic Programming (SQP) method, which shows that CKF is more accurate than EKF in estimating the SOC of lithium batteries. Ling et al. [23] proposed an estimation method based on the probability fusion of ACKF and AHCKF to further improve the estimation accuracy and robustness of SOC. However, although the method based on the ECM model can achieve better accuracy and robustness with relatively small calculation consumption, it is usually difficult to establish an accurate model because of the changes in the internal resistance and capacitance of the battery for different battery aging process and different temperatures in practical applications, and the OCV-SOC look-up tables will also be affected by different battery aging process and different temperatures, such error will increase if the update is not timely.

The data-driven method is not constrained by the physical relationship, which is easier to establish a high-efficient model. The non-linear relationship between SOC and measurable signals such as current, voltage and temperature can be established by training a substantial number of sample data. With appropriate training, higher estimation accuracy can be achieved [24]. At the same time, the battery SOC estimation model established based on the data-driven method has a strong generalization in different working environments of the battery. Chaoui et al. [25] realized the estimation of battery SOC through a recurrent neural network (RNN), but ordinary RNN cannot solve the long-term dependence problem. Chemali et al. [26] used the Long short-term memory (LSTM) to achieve the estimation of SOC with high accuracy, and the same model can be applied at different temperatures, overcoming the problem that model parameters need to be updated due to changes in the working conditions. One-dimensional convolution also has an excellent performance in solving time series problems. Hannan et al. [27] realized high-accuracy SOC estimation by using a one-dimensional depth full convolution network with a learning rate optimization

algorithm. In general, the data-driven method still has problems such as large computation consumption and limited accuracy. The data-driven method is significantly dependent on the quality and quantity of the data. When the specific working conditions of a specific battery are used as training data, because the training data cannot fully reflect the inherent characteristics of the battery, large estimation errors may occur in different battery systems and under the complex and dynamic working conditions of the battery. Meanwhile, when the amount of data is large, the training of data-driven models requires a huge computing cost.

Obviously, each individual method has its own drawbacks when using a single method to estimate the battery SOC. Collaboration between various approaches is an effective solution to obtain more accurate results. Tian et al. [28] extracted the open circuit voltage (OCV), ohmic response and polarization voltage from the voltage and current sequence as features and input them into the deep neural network (DNN) for battery SOC estimation. Feng et al. [29] fused the simplified single particle model (SSPM) with the thermal model to form a lumped model to estimate the battery SOC through the unscented Kalman filter (UKF), and used the feedforward neural network to improve the performance of the lumped model under an extreme working environment. Ragone et al. [30] combined a vehicle model, an electrochemical degradation model, a battery thermal model and a machine learning to estimate the SOC for the EVs. Although the method of model collaboration can achieve high accuracy, the constructed model has a high degree of complexity, which will increase calculation consumption and is unsuitable for vehicle applications. At the same time, many collaboration models still have the problem of generalization under different temperatures.

In summary, there are several main issues in the estimation of battery SOC: one is that the generalization of the estimation of SOC is weak under various working conditions, especially for temperature conditions, and the impact will become more critical with battery degradation and inconsistency. It is essential to update the model parameters in a timely manner to ensure accuracy under different temperatures, but in the data-driven method, the temperature can be simply regarded as a feature and directly input into the model for training and prediction, which greatly reduces the difficulty of the operation and effectively solves the problem of SOC estimation under different temperatures. Another problem is the contradiction between the constrained computing power of vehicle terminals in practical applications and the requirements for accurate algorithms functional in real-time. The high-accuracy results of most methods are usually obtained by building complex models. However, in practical applications, excessive computing consumption may lead to slow signal response time, which cannot guarantee the real-time performance of SOC estimation due to the constraint computing power of the onboard BMS. Although some methods also have high computing efficiency, these methods are all aimed at predicting battery cells and cell only. In practical applications, the electric vehicles are equipped with battery packs, which leads to the calculation consumption multiplied, where the problems mentioned above still stand.

In order to deal with the above problems, this paper proposes an end-cloud collaboration approach to online estimate SOC. The deep learning model can directly input the simple temperature as a feature into the network for training and prediction, which solves the problem of generalization under different temperatures, and the end-cloud collaborative working mode can share the calculation consumption, so as to solve the contradiction between accuracy and real-time. The deep learning model with large calculation consumption is deployed on the cloud-side and the real-time collected measurable signals are uploaded to the cloud for offline estimation. The data-driven model on the cloud-side is built with the CNN-LSTM network as the core. This model can effectively deal with the time series problem of SOC estimation in the battery, a non-linear system, and can obtain satisfactory estimation accuracy under different temperatures. The state function constructed by the coulomb counting method with less calculation consumption is deployed on the end-side, and the online calculation is carried out through the

real-time collected current. At the same time, the Kalman filtering algorithm is deployed on the end-side to fuse the estimation results of the cloud-side and end-side, to achieve high-accuracy and real-time SOC online estimation. The accuracy and robustness of the proposed end-cloud collaboration approach were evaluated with three dynamic driving conditions, including dynamic stress test (DST), US06 driving schedule and federal urban driving schedule (FUDS) under different temperatures and different initial errors.

This paper is organized as follows:

Section 2 shows the theory of the models and algorithms deployed on the cloud-side and end-side, as well as the structure of the whole end-cloud collaboration system. Section 3 describes the settings of software and hardware in the experiment and the evaluation index of the experiment. Section 4 describes the estimation results and discussions of the proposed method. Finally, Section 5 summarizes the main conclusions.

2. Methodology

The proposed end-cloud collaboration approach consists of three parts:

- (1) Deep learning model on the cloud-side, which has high accuracy, poor real-time performance and large computing consumption, and is suitable for cloud computing.
- (2) The state function constructed by the coulomb counting method on the end-side, whose accuracy is susceptible to the interference of initial value deviation, with poor accuracy and robustness, but the calculation cost is small, which is suitable for the arrangement of the end-side for calculation.
- (3) Fusion algorithm. Kalman’s filtering algorithm is used to combine the result of the end-side with that of the cloud-side to obtain more accurate results.

2.1. Data-Driven Model on Cloud

The deep learning model on the cloud-side used in this paper is constructed with the CNN-LSTM model as the base.

2.1.1. CNN-LSTM Introduction

CNN and LSTM are two commonly used networks in deep learning. CNN is mainly composed of convolution layer and pooling layer, which can extract potential features from sample data through the convolution kernel, and there is an activation function after the convolution layer to introduce the nonlinear characteristics to the model [31]. CNN has the characteristics of weight sharing and local connection, which can significantly reduce the number of model parameters, speed up training and improve generalization performance. One-dimensional CNN is mainly used in timeseries problems and the structure is shown in Figure 1a. The one-dimensional convolution is calculated as:

$$x_k^l = f\left(\sum_{i=1}^N x_i^{l-1} * w_{ik}^l + b_k^l\right) \tag{1}$$

where $*$ is the convolution operation, x_k^l is the k -th convolution map of the layer l , f is the activation function, N is the number of convolution maps, w_{ik}^l is the weight of the operation i of the convolution kernel k of layer l and b_k^l is the offset of the convolution kernel k corresponding to the layer l .

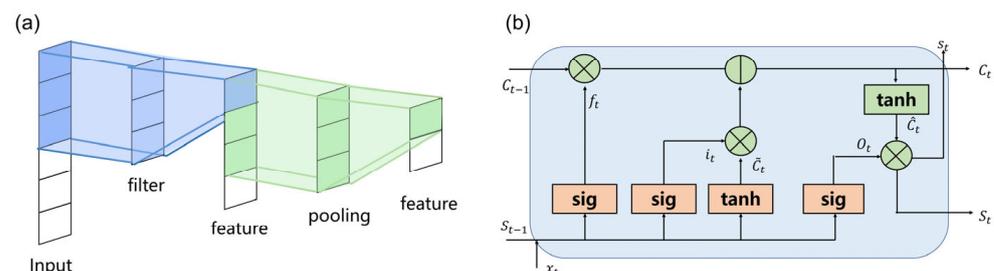


Figure 1. Schematic of LSTM and CNN. (a) Schematic of 1d-CNN. (b) Schematic of an LSTM unit.

LSTM is a variant of RNN, which adds input gates, output gates and forgetting gates to the hidden layer, and adds cells for memory storage, which can process the time series problem of long-term dependence [32]. The schematic of a unit is shown in Figure 1b. Each unit of LSTM can be expressed as follows:

Firstly, the LSTM units receive the x_j and h_{j-1} and the forget gate is used to control which information to be forgotten in the cell state:

$$f_j = \sigma(W_f \cdot [h_{j-1}, x_j] + b_f) \quad (2)$$

where sigmoid is a nonlinear activation function, and will limit the value to the range of 0 to 1, which determines the proportion of forgotten information and update the unit status ' c_j '.

Secondly, the input gate processes the input data of the current sequence position and determines which information to be updated to update the cell state. The input gate includes two parts: one uses the sigmoid activation function to determine which new information in the input is added to the cell state, and after determining the reserved input new information, the tanh activation function is used to generate new candidate vectors, and then input the information to be reserved into the cell state. The input gate can be described as follows:

$$i_j = \sigma(W_i \cdot [h_{j-1}, x_j] + b_i) \quad (3)$$

$$g_j = \tanh(W_g [h_{j-1}, x_j] + b_j) \quad (4)$$

Next, after determining the output of the forgetting gate and the input gate, the unit state update from c_{j-1} to c_j :

$$c_j = c_{j-1} * f_j + i_j * g_j \quad (5)$$

where $c_{j-1} * f_j$ is the information that to be retained $i_j * g_j$ is the new information to add. The sum of the two is the cell state of this sequence.

Finally, using the sigmoid activation function to determine which part of the content needs to be output, then use the tanh activation function to process the cell state, and then multiply the two parts to obtain the final output:

$$o_j = \sigma(W_o \cdot [h_{j-1}, x_j] + b_o) \quad (6)$$

$$h_j = o_j * \tanh(c_j) \quad (7)$$

where h_j is the new hidden state, and the c_j is the unit state.

SOC estimation is a typical time series estimation problem but it is aimed at highly complex dynamic conditions and requires high real-time, which causes the difficulty to further extract features from the data. CNN can extract the potential features from sample data, while LSTM can capture the long-term dependence in time series problems, which has great advantages in processing time series data. Therefore, we use a hybrid model that adopts LSTM as the backend of CNN, which is called CNN-LSTM, in which CNN is used to extract features from data and LSTM is used to learn the time dependency relationship.

2.1.2. Structure of Deep Learning Model

The model is constructed with the input layer, Conv1d layer, Maxpooling layer, LSTM layer and full connection layer. Since we are dealing with one-dimensional time series problems, one-dimensional convolution is used for the convolution layer. The architecture of the CNN-LSTM is shown in Figure 2. The current, voltage and temperature are used as the input. Since the input structure required by 1d-CNN-LSTM is a 3d-tensor, the input data is built into a time series format. First, build a time series according to a certain time series length N . The input and the output structures are "many to one". Considering that the transmission delay in practical applications will affect the real-time performance, this work focuses on the SOC estimation of the future time step, that is, extrapolating the SOC

of the future time based on historical data. The input and output of the CNN-LSTM can be described as follows:

$$x_k = \begin{bmatrix} I_{k-N+1} & V_{k-N+1} & T_{k-N+1} \\ I_{k-N+2} & V_{k-N+2} & T_{k-N+2} \\ \dots & \dots & \dots \\ I_k & V_k & T_k \end{bmatrix} \tag{8}$$

$$y_k = SOC_{k+l} \tag{9}$$

where I, V and T are the current, voltage and temperature, respectively. SOC_{k+l} is the SOC at time $k + l$, where l is the extrapolation step size and is selected as 1 in this paper. x_k is the input matrix of CNN-LSTM and y_k is the output of CNN-LSTM at time k .

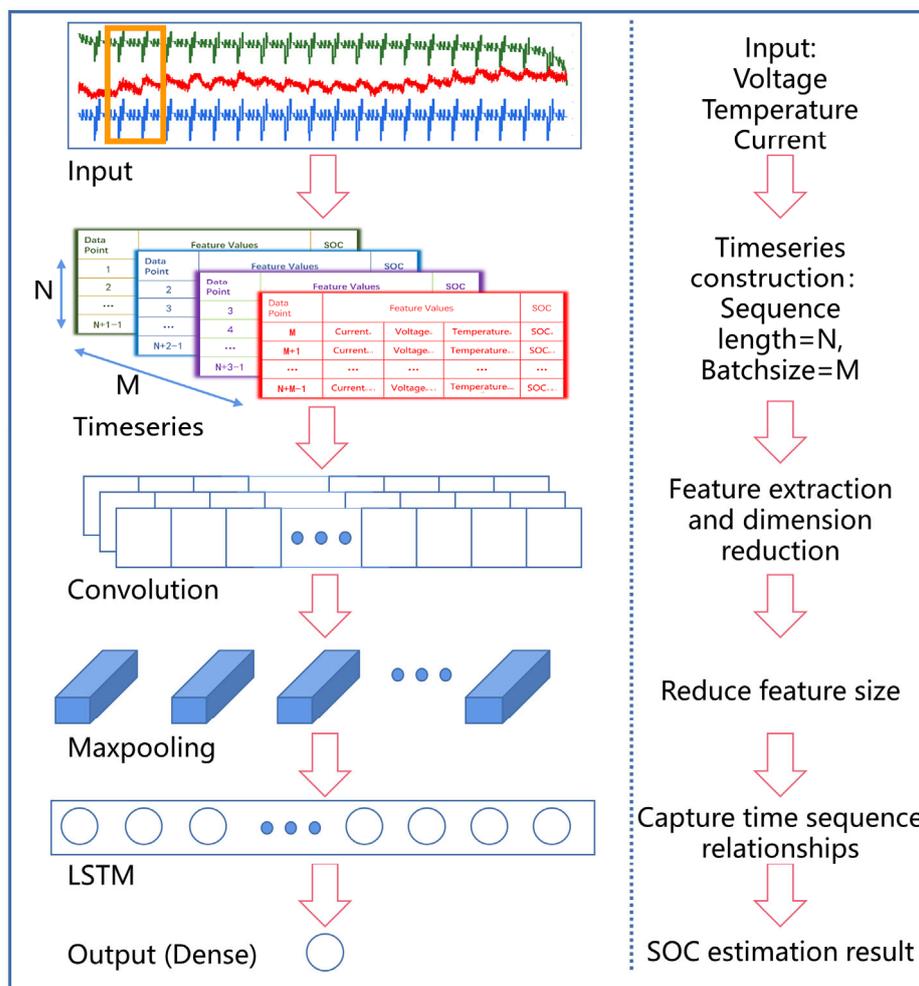


Figure 2. The architecture of the CNN-LSTM.

Generally, a longer time series length will improve the accuracy of the estimation but will reduce the efficiency of the model. For the sake of real-time, it is not appropriate to choose too large a time series. After comprehensive consideration of model accuracy and model efficiency and debugging, the time series length is selected as 300. The constructed time series is constructed into a batch according to the batch size subsequently. With the increase of the batch size, the training speed will be accelerated, but model will converge to the local optimal point easily. After comprehensive consideration of the model efficiency and prediction accuracy, the batch size is selected as 2560. Then, the input data is processed through CNN; with the operation of convolution and pooling, CNN extracts features from input and reduces the feature dimension. The convolution layer extracts potential features from the original data through the sliding of the filter in the time dimension. The pooling

layer uses maxpooling, and the processing of the pooling layer can further reduce the feature size to reduce the amount of computation. After that, the LSTM receives the data processed by CNN and the LSTM network adjust the parameters in each gate through training so that it can learn the time-dependence relationship between data from features extracted by the CNN, thus effectively solving the time series prediction problems. Finally, the predicted value is output by the full connection layer, and the sigmoid activation function is added to the full connection layer. The detailed training and testing process can be found in Figure S1 and Table S1.

2.2. CCM in End

The state function constructed by CCM is deployed at the end-side. The change in charge is obtained through integrating the current with time, the coulomb counting method is calculated as:

$$SOC = SOC_0 - \frac{1}{C_N} \int_0^t \eta I d\tau \quad (10)$$

where C_N is the rated capacity of the battery, I is the current and η is the charge and discharge efficiency.

When the CCM is carried in vehicle BMS, it needs to be discretized first. In addition, the integration of the cloud-side model and the end-side also needs the discrete form, which can be described as follows:

$$SOC_k = SOC_{k-1} - \frac{I_{k-1} \times \Delta t}{C_N} \quad (11)$$

where C_N is the time interval between time k and time $k - 1$, I_{k-1} is the current at time $k - 1$, SOC_k is the SOC at time k obtained by the CCM and SOC_{k-1} is the SOC at time $k - 1$ obtained through the CCM. This method has a clear physical basis and simple calculation, and consumes little computational resources. However, as an open-loop method, it is easy to encounter the influence of cumulative error, so it is necessary to use certain methods to correct it in practical application.

2.3. End-Cloud Collaboration

Kalman's filter algorithm is deployed on the end-side for fusion. The KF algorithm estimates the system state based on the linear system state function through the input and output observation data of the system, but only linear systems can adopt the KF algorithm. The UKF algorithm is an improvement on the KF algorithm, which adopts the Kalman linear filtering framework and approximates the probability density of nonlinear functions based on the Bayesian transform and UT transform. The UKF algorithm has high accuracy in nonlinear problems, and can also be applied to linear problems. Although the state function and measurement function constructed in the application of this paper are all linear functions, considering that the follow-up work will be extended to the joint estimation of multi-state quantities, in order to deal with the possible nonlinear problems, this paper will adopt the KF and UKF algorithms, respectively, as collaborative algorithms, and make a comparison between these two algorithms, to explore the impact of the approximate transformation of the UKF on the accuracy.

2.3.1. KF Algorithm

The Kalman filter is a filtering algorithm based on optimal estimation [33]. By comprehensively considering estimated values and measured values, it fuses state variables and observation variables to get a result close to real value, which will as state variables of the next process, and then fuses with the observation variables of the next process, so iteratively. The KF algorithm can be described as follows:

$$X_k = AX_{k-1} + Bu_k + w_k \quad (12)$$

$$Z_k = HX_k + v_k \tag{13}$$

Expression (12) is a state function, where X_k represents the current state, X_{k-1} represents the previous state, u_k represents the input, w_k represents the process noise, A and B represent the state transition matrix. Expression (13) is a measurement function, where Z_k represents measured variables, v_k represents measurement noise and H represents the transfer parameters matrix of the measured variables. The working process of Kalman filter is as follows:

time update:

$$\hat{x}_k^- = A\hat{x}_k + Bu_{k-1} \tag{14}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{15}$$

measurement update:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{16}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{17}$$

$$P_k = (I - K_k H)P_k^- \tag{18}$$

where \hat{x}_k^- is the prior estimate of the state, \hat{x}_k is the posteriori optimal estimate, P_k^- is the prior estimates of covariance, P_k is the posterior optimal estimation of the covariance, K_k is the Kalman gain and I is a complete identity matrix.

2.3.2. UKF Algorithm

The UKF is a method that enables nonlinear problems to be applied to the KF system only applicable to linear problems through UT transformation [34]. The UKF algorithm can be described as follows:

$$x_k = Ax_{k-1} + Bu_k + \omega_k \tag{19}$$

$$y_k = Cx_{k-1} + Du_k + v_k \tag{20}$$

where, ω_k is the system noise, and v_k is the measurement noise.

The process of UKF algorithm is as follows:

First determine the initial value of the status value x_0 and the initial covariance of the status error P_0 , and then calculate the sampling points:

$$x_k^0 = \hat{x}_k \tag{21}$$

$$x_k^i = \hat{x}_k + \left(\sqrt{(L + \lambda)P_{k-1}} \right)_{(i)}, i = 1, 2 \dots L \tag{22}$$

$$x_k^i = \hat{x}_k - \left(\sqrt{(L + \lambda)P_{k-1}} \right)_{(i)}, i = L + 1, L + 2, \dots 2L \tag{23}$$

The weight value is calculated as follows:

$$\lambda = \alpha^2(L + k_i) - L \tag{24}$$

$$W_m^0 = \frac{\lambda}{L + \lambda}, W_m^i = \frac{1}{2(L + \lambda)}, i = 1, 2 \dots 2L \tag{25}$$

$$W_c^0 = \frac{\lambda}{L + \lambda} + 1 - \alpha^2 + \beta, W_c^i = \frac{1}{2(L + \lambda)}, i = 1, 2, \dots 2L \tag{26}$$

where λ is the scale factor between the mean of random variable x and the sigma sampling points, $\left(\sqrt{(L + \lambda)P_{k-1}} \right)_{(i)}$ is the row i or column i of the square root matrix obtained by

Cholesky decomposition of matrix $(L + \lambda)P_{k-1}$, α and k_i are the scale factor, and β is a high-order characteristic reflecting the state history information.

Then update the prior status value \bar{x}_{k+1} and system variance prediction value P_{xx} :

$$\bar{x}_{k+1} = \sum_{i=0}^{2L} W_m^i x_k^i \quad (27)$$

$$P_{xx} = \sum_{i=0}^{2L} \left(W_c^i (x_k^i - x_{k+1}) (x_k^i - x_{k+1})^T \right) + Q_k \quad (28)$$

where Q_k is the system noise covariance matrix. Next, update the observations \hat{y}_{k+1} and predicted value of observation variance P_{yy} .

$$\hat{y}_{k+1} = \sum_{i=0}^{2L} W_m^i y_k^i \quad (29)$$

$$P_{yy} = \sum_{i=0}^{2L} \left(W_c^i (y_k^i - \hat{y}_{k+1}) (y_k^i - \hat{y}_{k+1})^T \right) + R_k \quad (30)$$

Finally, update the covariance P_{xy} , the posterior state value \hat{x}_{k+1} and the posterior state error covariance P_k .

$$P_{xy} = \sum_{i=0}^{2L} W_c^i (x_k^i - x_{k+1}) (y_k^i - \hat{y}_{k+1})^T \quad (31)$$

$$K_k = \frac{P_{xy}}{P_{yy}} \quad (32)$$

$$\hat{x}_{k+1} = x_{k+1} + K_k (y_{k+1} - \hat{y}_{k+1}) \quad (33)$$

$$P_k = P_{xx} - K_k P_{xy}^T \quad (34)$$

2.3.3. Collaboration Process

For the SOC estimation, the state vector is $x = \text{SOC}$, which follows the coulomb counting method to construct the state function and deploy on the end-side. The measurement vector is the SOC outputs by the CNN-LSTM network. The CNN-LSTM network deployed on the cloud-side is applied as the measurement function. The Kalman's filter is deployed on the end-side, which receives the results from both sides of the end and cloud and performs result fusion. Therefore, a state space model is established as follows:

$$\text{SOC}_k = \text{SOC}_{k-1} - \frac{I_{k-1} \times \Delta t}{C_n} + \omega \quad (35)$$

$$\text{LSTM}_k = \text{SOC}_k + v \quad (36)$$

where I_k is the current at time k , C_n is the nominal capacity, $\omega \sim N(0, Q)$ is the Gaussian state noise and $v \sim N(0, R)$ is the Gaussian measurement noise.

The framework of the proposed end-cloud collaboration SOC estimation system is represented in Figure 3; the whole system is constructed based on the CHAIN framework [35,36]. On the cloud-side, the deep learning model estimates SOC, and the estimated results are regarded as the measured value. On the end-side, the state estimation is performed and the calculation results are regarded as the observed values, and then the collaborative algorithm is used to combine the calculation results of the end-side and the cloud-side, and update the result at each time step to obtain more accurate results. Specifically, in the training process, the deep learning model on the cloud-side uses a substantial amount of offline data from public datasets, enterprise databases, online data storage and other aspects to complete the training. During the estimation process, the voltage, current and temperature collected from the end-side are uploaded to the cloud-side. To estimate

the battery SOC, and the cloud-side offline high-accuracy estimation results are obtained subsequently, the deep learning model on the cloud-side can estimate the SOC of the battery in the form of extrapolated estimation of historical data and the result will be used as the measured values of the Kalman’s filter, that is, the z_k in Expression (17) and the y_{k+1} in Expression (33) to update the end results. The coulomb counting method in end-side uses the current collected from the end-side to calculate the SOC, and the result will be used as the observed value of the Kalman’s filter, that is, the \hat{x}_k^- in Expression (14) and the \hat{x}_k in Expression (24). The Kalman’s filter is deployed on the end-side, and the real-time estimation results from the end-side and the estimation results from the cloud-side are received at the same time. Because the cloud-side is estimated by extrapolation, the cloud-side estimation results of the current time step have been obtained and transmitted in the previous time, so it can ensure that the transmission delay will not affect the real-time performance. The Kalman’s filter fuses the results from both sides to obtain the end-cloud collaborative high-accuracy and real-time estimation result, that is, the \hat{x}_k in Expression (17) and the \hat{x}_{k+1} in Expression (33), which is the final output of the entire end-cloud collaboration system. In practical applications, the whole end-cloud collaboration process can be realized through a cloud-based BMS [37].

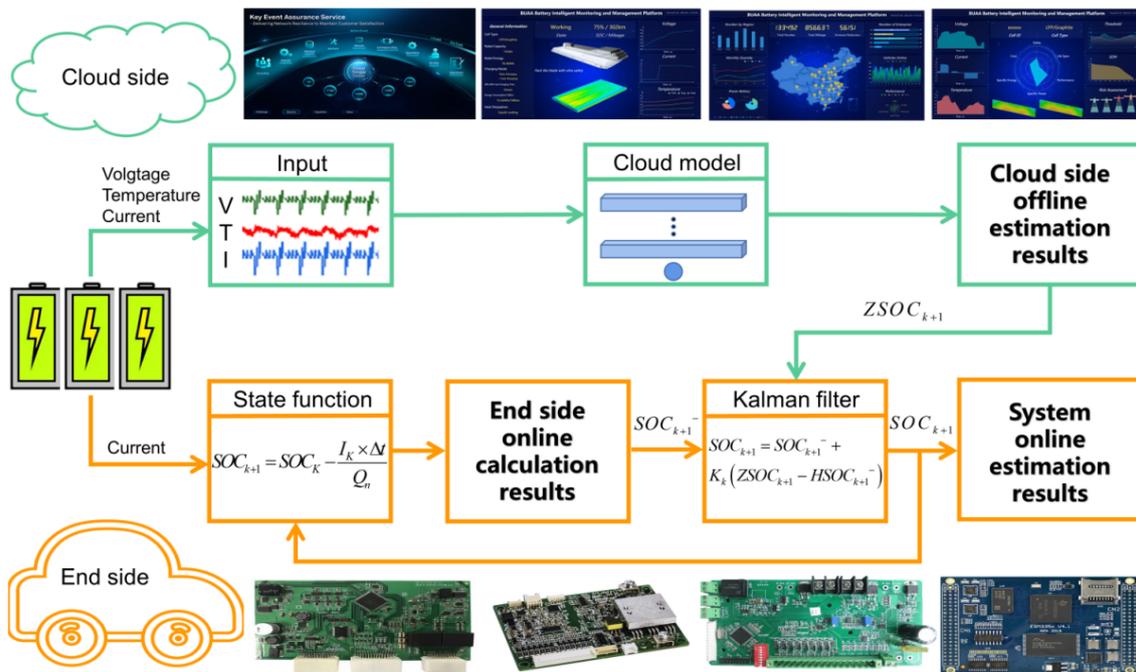


Figure 3. Framework of the proposed end-cloud collaboration SOC estimation system based on CHAIN.

3. Experimental Setup

3.1. Dataset Description

Three dynamic driving profiles are employed to evaluate the proposed method, including the DST, US06 and FUDS. The specific details of each profile can be found in the USABC manuals [38]. The experimental data comes from a public dataset from the data repository of the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland [39]. In the experiment, the cylindrical 18650 LiFePO4 batteries were tested under different temperatures of 0 °C, 10 °C, 25 °C, 30 °C, 40 °C and 50 °C, the specifications of the batteries are shown in Table 1, and the voltage, current, temperature and SOC profiles under 25 °C are shown in Figure 4. More details about the dataset are shown in Figures S2–S4. Obviously, there are significant differences between these three driving conditions. To verify the generalization performance of the proposed method, the training set is constructed using the data of DST profile, and the test set is constructed using the data

of US06 and FUDS profiles. Before training and testing, the datasets are standardized to optimize the training process and reduce the difference in the order of magnitude between the data. The data have been standardized as follows:

$$x_{sta} = \frac{x - \bar{x}}{\sigma} \tag{37}$$

where x is the original value, \bar{x} is the mean value and σ is the standard deviation value.

Table 1. Specifications of the batteries.

Parameters	Specifications
Type	A123 18650
Cathode	LiFePO ₄
Anode	Graphite
Nominal capacity	1.1 Ah
Charge cut-off voltage	3.6 V
Discharge cut-off voltage	2.0 V
End-of-charge current	0.01 C

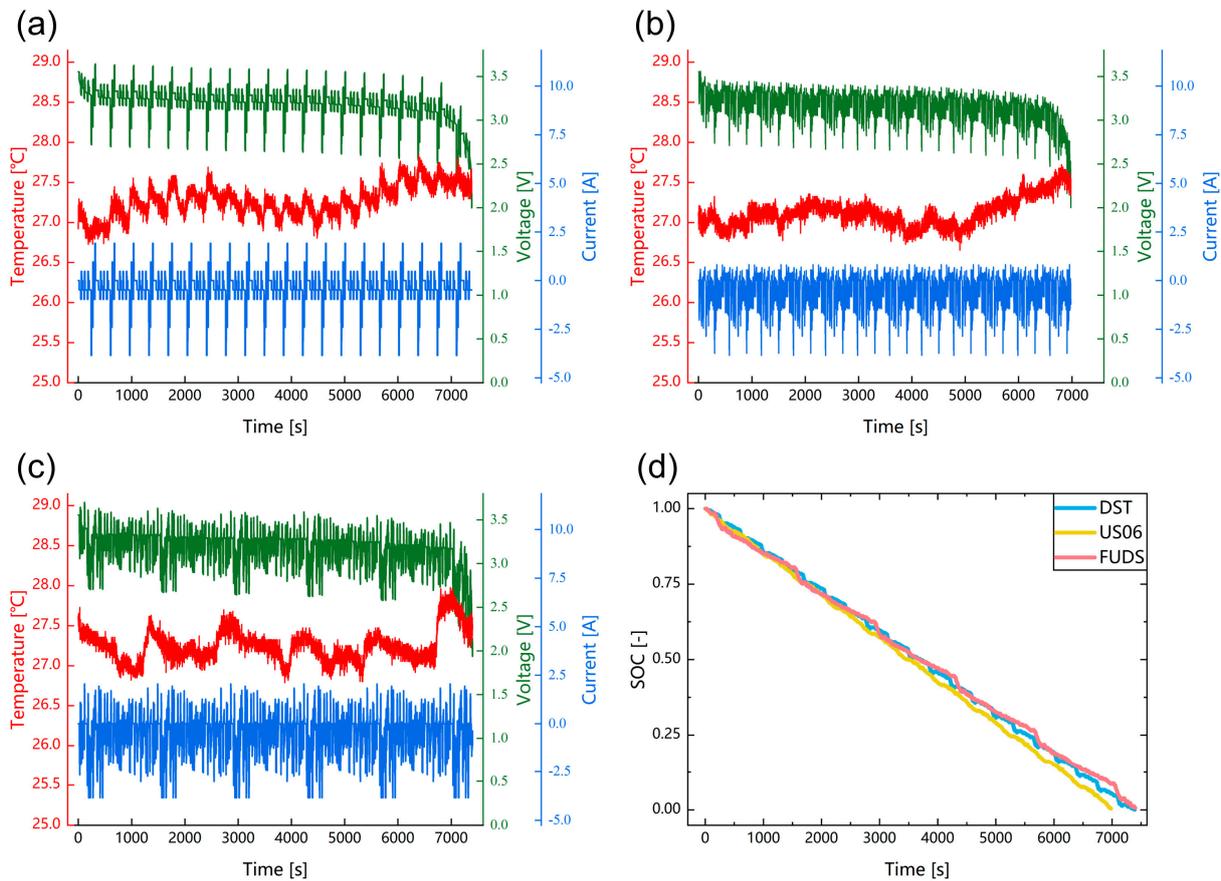


Figure 4. Current profiles and measured voltage and temperature under different driving cycles at 25 °C for (a) DST, (b) US06 and (c) FUDS; (d) donates the SOC estimation for three driving cycles.

3.2. Simulation Setup

The Bayesian optimization and the RMSprop algorithm were used to improve the training efficiency of the model. The RMSprop algorithm is an effective and practical deep learning model optimization algorithm which has been proven in practice, and it is one of the most advanced and frequently used optimization algorithms can efficiently improve the convergence speed and accuracy.

The Bayesian optimization method is employed for hyper-parameter optimization [40]. The hyper-parameters are parameters preset before training, rather than obtained by training. The performance of the deep learning model is highly dependent on the hyper-parameters. However, the values of the hyper-parameters are usually unable to be applied for batteries with different material systems and different driving conditions. The method of manually setting hyper-parameters is highly dependent on parameter adjustment experience and requires a lot of debugging, which will greatly reduce the training efficiency, especially in applications with large data volumes and complex model structures. The Bayesian optimization methods could find the optimal hyper-parameters automatically to obtain the best result of the deep learning model. In this paper, the size and number of filters in convolution layers, the learning rate and the number of units for the LSTM are optimized through this method. The calculation of the Bayesian optimization method could be represented as follows:

$$d^* = \operatorname{argmin} J(d), D \in (a, b), (d \in D) \quad (38)$$

where d is the hyper-parameter value, d^* is the optimal value.

In this paper, the whole network has been built based on TensorFlow, and the training process is accelerated through the compute unified device architecture (CUDA). All experiments are conducted on a server with Intel core i7 CPU and NVIDIA GTX 3090 GPU.

3.3. Evaluation Index

The proposed end-cloud collaboration approach is evaluated with the mean absolute error (MAE) and root mean square error (RMSE) criteria as the evaluation index:

$$\text{MAE} = \frac{1}{N} \sum_{k=1}^N |y_k - y_k^*| \quad (39)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{k=1}^N (y_k - y_k^*)^2} \quad (40)$$

where y_k represents the real SOC and y_k^* the estimated SOC.

4. Result and Discussion

In this section, the accuracy and generalization of the CNN-LSTM network and the proposed end-cloud collaboration approach are evaluated firstly by estimating the SOC at different temperatures and different driving profiles. Secondly, the robustness of the proposed method is evaluated by manually setting different initial errors to estimate SOC under different temperatures. Meanwhile, the estimation results of KF and UKF as collaborative algorithms are compared in all experiments about the end-cloud collaboration approach.

4.1. Estimation Result of CNN-LSTM Network

In this subsection, the accuracy and generalization of the deep learning model on the cloud-side are evaluated at 0 °C, 25 °C and 40 °C, respectively. Figure 5a–f shows the estimation results and estimation errors of the CNN-LSTM network under US06 and FUDS profiles, respectively, while Figure 5g,h show all estimation errors in the form of histograms. Each subgraph contains one dotted line and two solid lines. The black solid line illustrates the real SOC, the green solid line illustrates the estimation results of the CNN-LSTM network and the dotted line illustrates the estimation error of the CNN-LSTM network. The MAE and RMSE of the estimation results are about 2% and 3%, respectively. In particular, at 25 °C, the MAE and RMSE of the FUDS profile are 1.44% and 1.94%, respectively. However, the estimation results of the CNN-LSTM network fluctuate significantly, with a maximum estimation error is about 10%, and all appear in the middle regions, because the voltage characteristic curve having a plateau section in the middle region of SOC during battery discharge and cause the voltage change slope is so small that enlarge the deviation of the

entire estimation results easily. Meanwhile, although the CNN-LSTM network has a strong generalization of the battery SOC estimation at different temperatures, the temperature will affect the estimation results. The error and fluctuation of the estimation results will increase at 0 °C and 40 °C. The maximum error is 8.57% at 25 °C under the US06 profile but will increase to 10% and 10.9%, respectively at 0 °C and 40 °C. At 25 °C under the FUDS profile, the maximum error is 7.8%, when the temperature is 0 °C and 40 °C, it will increase to 16.5% and 15.4%, respectively. The estimation error will also have a difference of about 1.5%. This is because the LiFeO₄ battery has a section of platform in the voltage characteristic curve and it will become flatter with the increase in temperature. Therefore, when the voltage measurement errors are the same, the estimation error of SOC will increase at high temperatures, while at low temperatures, with the increase in temperature, the internal resistance will increase rapidly and as a result, the polarization characteristic time at low temperature is very short, and the polarization voltage changes rapidly with temperature, increasing the estimation error.

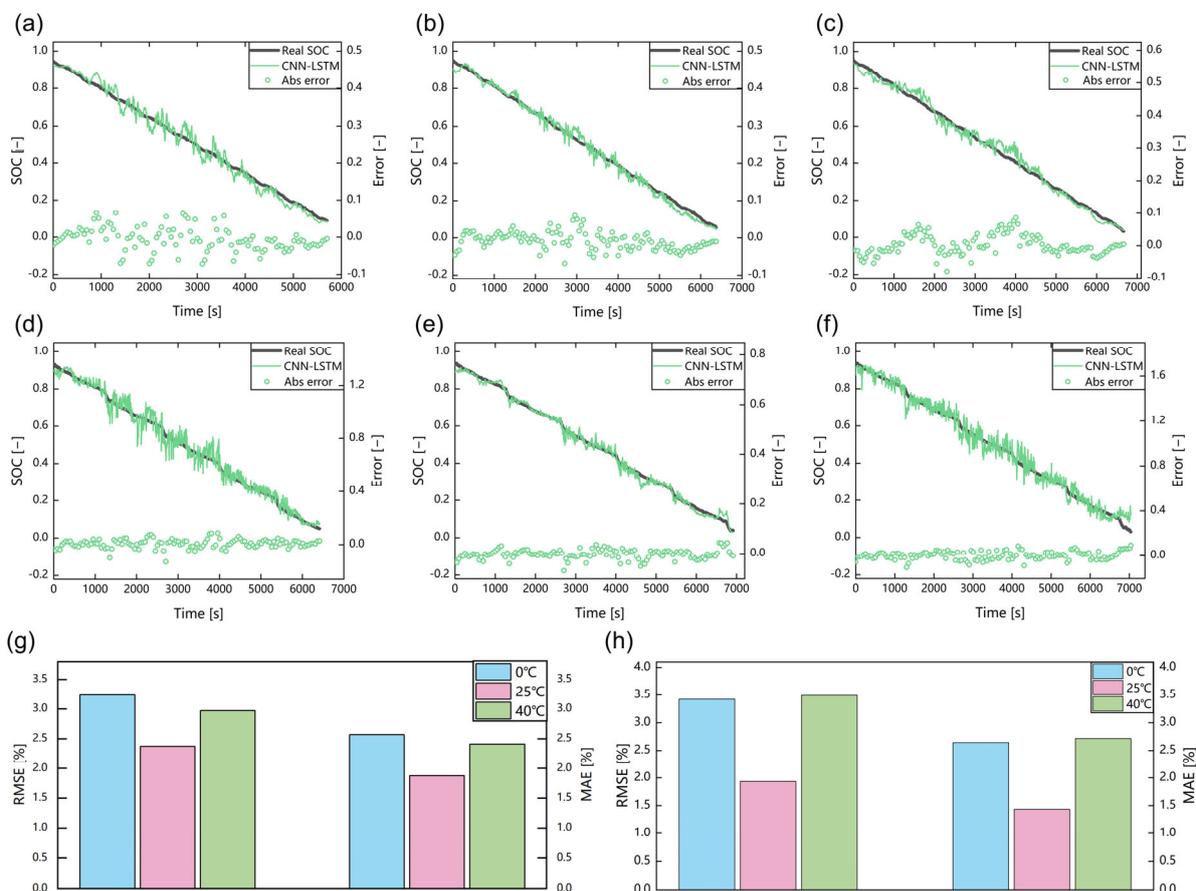


Figure 5. SOC estimation results with CNN-LSTM under different temperatures: (a–c) donate the estimation results for US06 driving cycle under 0 °C, 25 °C and 40 °C, respectively; (d–f) donate the estimation results for FUDS driving cycle under 0 °C, 25 °C and 40 °C, respectively; (g,h) donate the RMSE and MAE analysis for US06 driving cycle and FUDS driving cycle, respectively.

4.2. Estimation Result of End-Cloud Collaboration Approach under Different Temperatures

In this subsection, the estimation accuracy and generalization of the proposed end-cloud collaboration approach are evaluated at 0 °C, 25 °C and 40 °C, respectively. Figure 6a–f shows the estimation results and estimation errors of the end-cloud collaboration approach under US06 and FUDS profiles respectively, while Figure 6g,h show all estimation errors in the form of histograms. Each subgraph contains two dotted lines and three solid lines. The black solid line represents the real SOC, the blue solid line represents the estimation result

collaborated by KF, and the red solid line represents the estimation result collaborated by UKF. The dotted line represents the estimation error, and different colours correspond to the above solid lines one by one. The MAE and RMSE are all about 1%. In particular, at the FUDS profile under 40 °C, the MAE and RMSE are 0.58% and 0.77%, respectively. The accuracy of adopting KF and UKF as collaborative algorithms are almost the same, and the difference between MAE and RMSE is only about 0.01%. In most cases, the estimation accuracy of UKF is slightly lower than that of KF, but the difference is very small, which does not affect its application in practical applications. The difference between these two methods is little for the application is linear, but the approximate transformation will slightly reduce the accuracy when using UKF in linear problems. Table 2 summarizes the estimation error of the end-cloud collaboration approach and the CNN-LSTM network. It can be seen that after the integration of the end-cloud collaboration system, the MAE and RMSE are reduced by about 1% and 2%, respectively compared with those of the single data-driven model, and the RMSE is reduced by 2.73% under the FUDS profile, particularly at 40 °C. Meanwhile, the maximum error of the end-cloud collaboration system has also been greatly reduced, and the maximum error is within 5%, which is more stable than the data-driven model. Furthermore, the estimation results of the single data-driven model will be affected for temperature, while the end-cloud collaboration approach is more stable at different temperatures, and the estimation error at room temperature is only 0.2% different from that at 0 °C and 40 °C. It can be seen that the end-cloud collaboration approach proposed in this work has high accuracy, and strong generalization and stability under different temperatures and driving cycles.

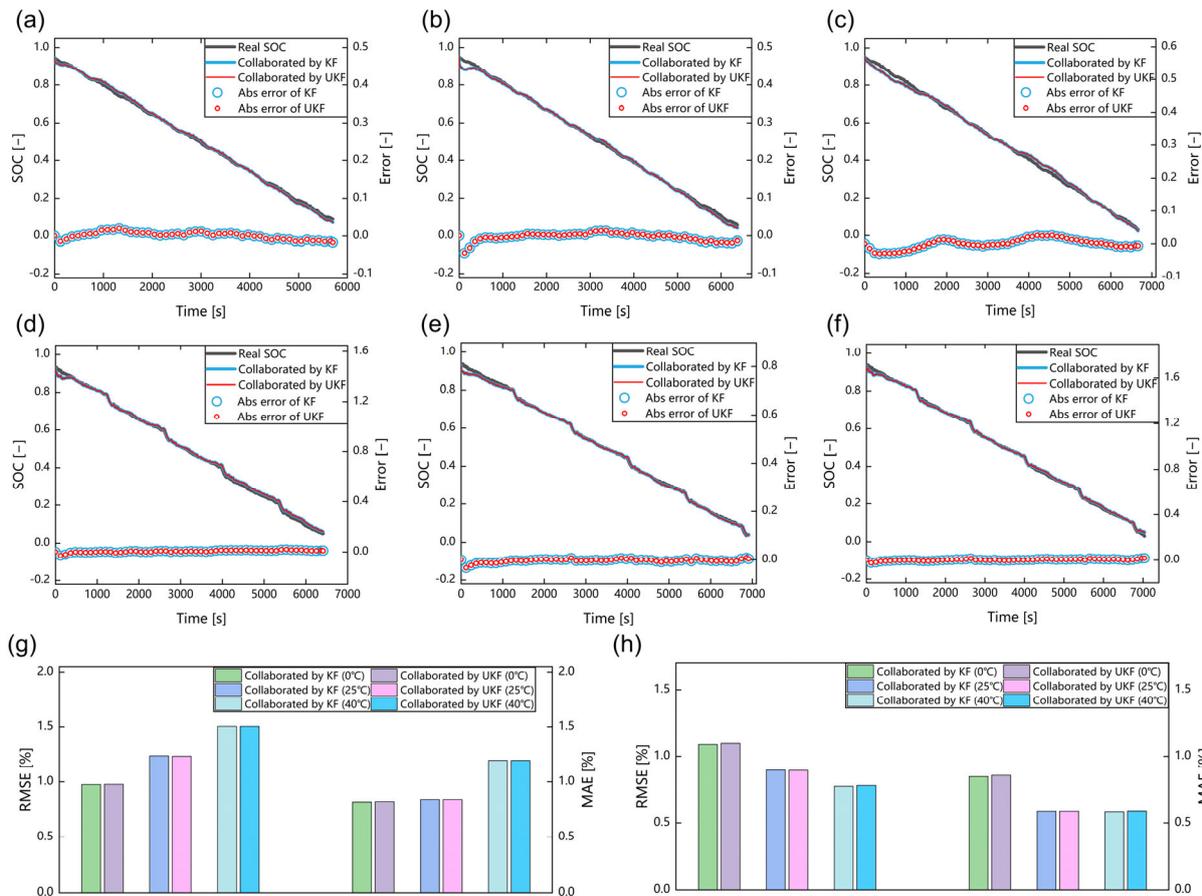


Figure 6. SOC estimation results with end-cloud collaboration system under different temperatures: (a–c) donate the estimation results for US06 driving cycle under 0 °C, 25 °C and 40 °C, respectively; (d–f) donate the estimation results for FUDS driving cycle under 0 °C, 25 °C and 40 °C, respectively; (g,h) donate the RMSE and MAE analysis for US06 driving cycle and FUDS driving cycle, respectively.

Table 2. RMSEs and MAEs of SOC estimation results with CNN-LSTM and end-cloud collaboration system under different temperatures.

Driving Cycle	Method	Evaluation Index	Estimation Error (%)		
			0 °C	25 °C	40 °C
US06	CNN-LSTM	MAE	2.57	1.88	2.41
		RMSE	3.24	2.38	2.98
	Collaborated by KF	MAE	0.82	0.84	1.20
		RMSE	0.98	1.24	1.50
	Collaborated by UKF	MAE	0.82	0.84	1.20
		RMSE	0.98	1.24	1.50
FUDS	CNN-LSTM	MAE	2.64	1.44	2.71
		RMSE	3.42	1.94	3.49
	Collaborated by KF	MAE	0.84	0.59	0.58
		RMSE	1.09	0.89	0.77
	Collaborated by UKF	MAE	0.86	0.86	0.59
		RMSE	1.10	0.89	0.78

4.3. Estimation Result of End-Cloud Collaboration Approach under Different Initial SOC Errors

In practical applications, the challenging point in estimating the battery SOC is about the SOC estimation with the consideration of varying model parameters, which change with the temperature and aging state of the battery. On the end-side, the state function is built by the CCM, whose accuracy is highly dependent on the initial error, which will change at different temperatures and different aging process. On the cloud-side, the deep learning model can directly input the temperature as a feature into the network for training and prediction. Meanwhile, training data with different data distributions can be used to train battery SOC estimation models under different aging states to deal with the impact of different aging process. Thus, the estimated results of the model on the cloud-side can correct the influence of varying working environment. Therefore, to verify the ability of the proposed end-cloud collaboration approach to resist initial value dependence and converge, in this subsection, taking the FUDS profile as an example, SOC is estimated by manually setting the initial value error at different temperatures. Figures 7a–c and 8a–c shows the estimation results and estimation errors of the end-cloud collaboration approach under the FUDS profile when the initial error is 20% and 40% of the true initial SOC, respectively. Figures 7d and 8d show all estimation errors in the form of histograms. The lines and colours in each subgraph represent the same content as described in Section 4.2. Specific error values are summarized in Table 3.

Observe the overall deviation. When the initial SOC is 80% of the real initial SOC, the estimation results of the end-cloud collaboration system can still guarantee high accuracy. Compared with the case without initial value error, MAE and RMSE only increased by about 0.02% and 0.1%, respectively. Under the harsh case with the initial SOC is 60% of the real initial SOC, MAE and RMSE are only increased by about 0.04% and 0.3%, respectively. In particular, under the US06 profile at 40 °C, when the initial SOC was 0.8 and 0.6 of the real initial SOC, RMSE increased by only 0.05% and 0.16%, respectively.

Meanwhile, observing the initial error correction, it can be seen that under different initial errors, the proposed end-cloud collaboration approach can quickly complete the initial error correction, so that the estimation results can quickly converge to the level under true condition. When the initial SOC is 80% of the real initial SOC, the estimation result can converge to the accuracy equivalent to that under the condition of accurate initial SOC within 30 s, and the error can be converged within the maximum error of the condition of accurate initial SOC about 10 s. When the initial SOC is 60% of the real initial SOC, it can converge to the same accuracy as the condition of accurate initial SOC within 50 s, and the error can converge within the maximum error under the condition of accurate initial SOC about 20 s. The convergence speed of KF and UKF is almost the same.

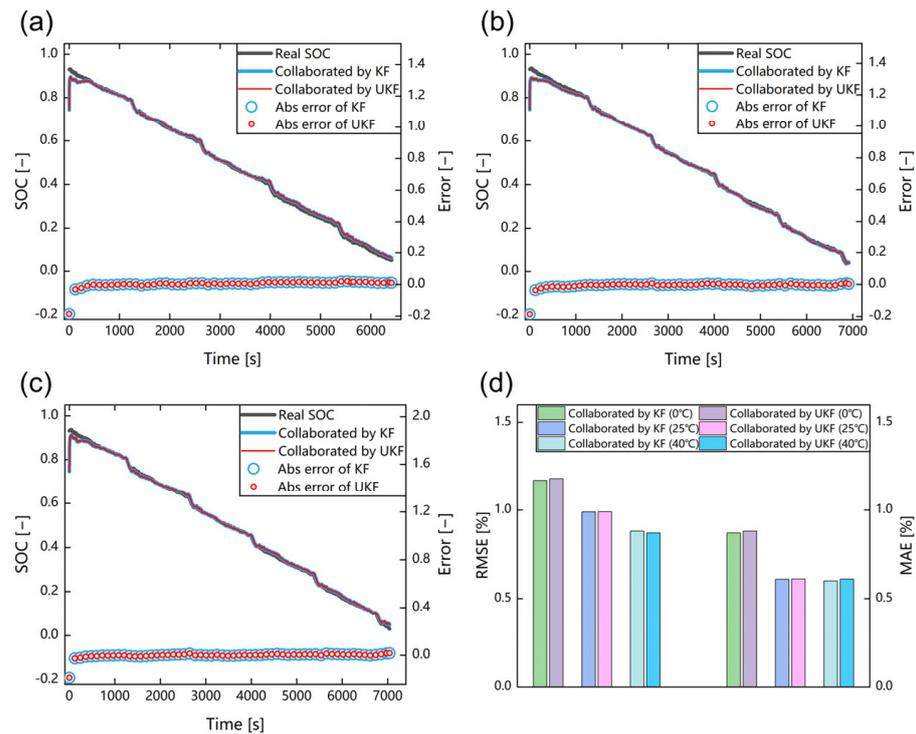


Figure 7. SOC estimation results with end-cloud collaboration system for FUDS driving cycle under different temperatures with an incorrect initial value of 80%. (a) Estimation result under 0 °C. (b) Estimation result under 25 °C. (c) Estimation result under 40 °C. (d) RMSE and MAE analysis for all estimation results.

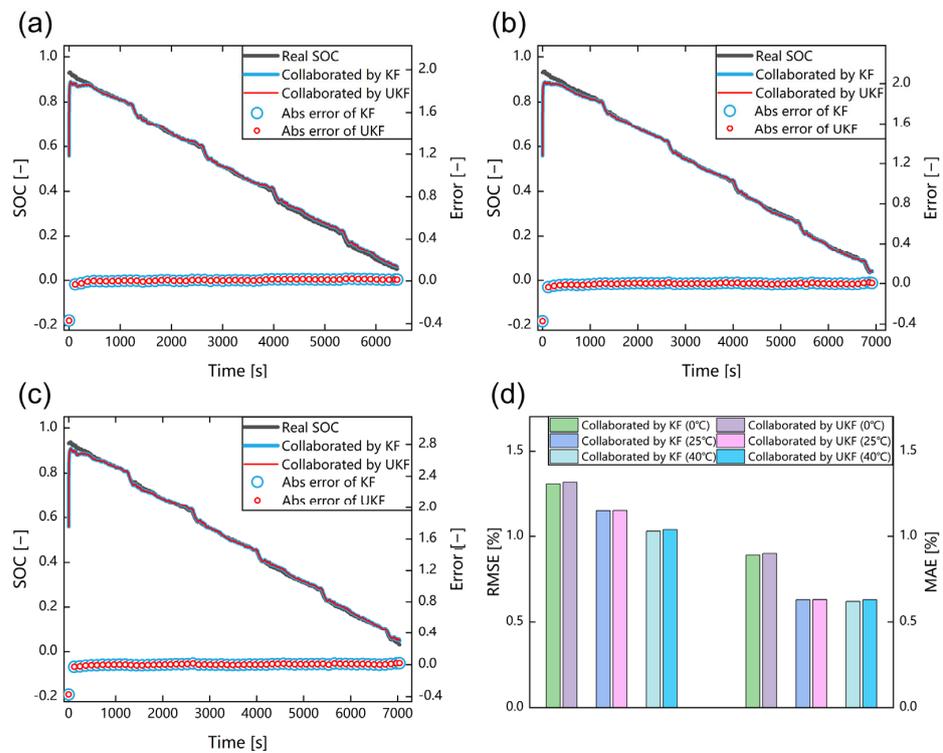


Figure 8. SOC estimation results with end-cloud collaboration system for FUDS driving cycle under different temperatures with an incorrect initial value of 60%. (a) Estimation result under 0 °C. (b) Estimation result under 25 °C. (c) Estimation result under 40 °C. (d) RMSE and MAE analysis for all estimation results.

Table 3. RMSEs and MAEs of SOC estimation results with end-cloud collaboration system under different temperatures with incorrect initial value of 80% and 60%.

Initial Value	Method	Evaluation Index	Estimation Error (%)		
			0 °C	25 °C	40 °C
80%	Collaborated by KF	MAE	0.87	0.61	0.60
		RMSE	1.17	0.99	0.88
	Collaborated by UKF	MAE	0.88	0.61	0.61
		RMSE	1.18	0.99	0.87
60%	Collaborated by KF	MAE	0.89	0.63	0.62
		RMSE	1.31	1.15	1.03
	Collaborated by UKF	MAE	0.90	0.63	0.63
		RMSE	1.32	1.15	1.04

In general, the proposed end-cloud collaboration approach can quickly complete the initial value error correction, thus avoiding the initial value dependence. Under different initial SOC conditions and different temperatures, it can achieve satisfactory accuracy, proving the robustness and stability of our method.

5. Conclusions

In this paper, the end-cloud collaboration approach is proposed to estimate the SOC for lithium-ion batteries online. The data-driven model is deployed on the cloud-side, and the coulomb counting method and Kalman's filter are deployed on the end-side. The data-driven model is built with the CNN-LSTM network as the core. The potential features in the original data are extracted through CNN, and the time series features are captured from the information extracted from the CNN network through LSTM subsequently. During the experiment, the Bayesian optimization algorithm is used for the automatic optimization of hyperparameters, and the training process is completed through a substantial number of data provided by public data sets or databases offline. The voltage, current and temperature collected from end-side and uploaded to cloud, where such data is used for estimation. The state function constructed by the CCM is deployed at the end-side and is calculated online through the current monitored in real-time. The estimation results of data-driven model are regarded as measured values, and calculation results of the coulomb counting method are regarded as observed values. The Kalman's filter simultaneously receives and fuses the calculation results from both the end side and cloud side, thus obtaining the final high-accuracy real-time output of the entire end-cloud collaboration system. Finally, the proposed end-cloud collaboration approach is evaluated under different temperatures and different initial errors with three dynamic driving profiles: DST, US06 and FUDS. The RMSE error is controlled less than 1.5%, and the maximum error is less than 5%, which indicates that the proposed end-cloud collaboration approach has high accuracy and strong robustness. By deploying a data-driven model on the cloud-side, deploying the coulomb counting method and Kalman's filter on the end-side, and fusing the calculation results of the end-side and the cloud-side through Kalman filtering algorithm to obtain high-accuracy estimation results, the amount of calculation is transferred through end-cloud collaboration, thus realizing high-accuracy, real-time online estimation of battery SOC. The data-driven model on cloud-side is constructed based on the CNN-LSTM network, the potential features in the original data are extracted through CNN, and the long-term dependence on the time series is captured from the feature information extracted from CNN through LSTM subsequently and iterative training is conducted for many times, thus conducting effective dynamic modelling. In the dynamic working condition, the features contained in the measurable signal and the long-term dependence relationship with SOC can be well extracted. This approach has strong generalization for it is based on the data-driven and CCM, which eliminates the necessity to adjust model parameters promptly and consult tables to build appropriate OCV-SOC relationships at different temperatures. Meanwhile, this method has strong robustness which can automatically correct the initial

value error to avoid initial value dependence and achieve satisfactory accuracy under different initial SOC conditions. It can complete the estimation task well under a variety of working conditions and initial errors and can be extended to batteries with different materials. This paper optimizes the estimation results and ensures real-time performance by arranging the models on the end-side and cloud-side respectively in the manner of end-cloud collaboration. Under the CHAIN framework, a high-accuracy and real-time SOC online estimation can be achieved through the proposed approach and this method can be extended to multi-state collaborative online estimation, making contributions to the development of the next generation of cloud-based BMS and end-cloud collaborative management system. In the future work, the impact of battery aging on SOC estimation will be considered through SOH and SOC collaborative estimation, and the proposed method will be further verified and optimized through real vehicle data verification and embedded development and can be expanded to application scenarios such as fault diagnosis and energy management.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/batteries9020114/s1>, Figure S1: The training and testing process of the CNN-LSTM; Table S1: The SOC estimation results with different network structures for US06 profiles at 25 °C. Figure S2: Current profiles and measured voltage and temperature under different driving cycles and different temperatures; Figure S3: Other variable values including charge capacity, discharge capacity, charge energy and discharge energy for the dataset under different driving cycles and different temperatures; Figure S4: Correlation analysis for the variable values with SOC.

Author Contributions: Conceptualization, S.Y., X.L. and W.W.; methodology, L.Z., W.W. and B.M.; software, W.W. and B.Z.; validation, W.W., L.Z. and H.Y.; investigation, B.M.; resources, X.L. and S.Y.; writing—original draft preparation, W.W.; writing—review and editing, X.H.; visualization, K.Y., B.M. and X.H.; supervision, S.Y. and X.L.; project administration, S.Y.; funding acquisition, S.Y. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Natural Science Foundation of China (No. 52102470).

Data Availability Statement: The data presented in this study are openly available in [CALCE Battery Research Group] at [<https://web.calce.umd.edu/batteries/data.htm>, accessed on 12 June 2022], reference number [39].

Acknowledgments: The authors would also like to acknowledge Shuimu Tsinghua Scholar Program for funding this work for Xiao Hua.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Or, T.; Gourley, S.W.D.; Kaliyappan, K.; Yu, A.; Chen, Z. Recycling of Mixed Cathode Lithium-Ion Batteries for Electric Vehicles: Current Status and Future Outlook. *Carbon Energy* **2020**, *2*, 6–43. [[CrossRef](#)]
2. Wu, J.; Cao, Y.; Zhao, H.; Mao, J.; Guo, Z. The Critical Role of Carbon in Marrying Silicon and Graphite Anodes for High-Energy Lithium-Ion Batteries. *Carbon Energy* **2019**, *1*, 57–76. [[CrossRef](#)]
3. Liu, X.; Lai, W.; Peng, J.; Gao, Y.; Zhang, H.; Yang, Z.; He, X.; Hu, Z.; Li, L.; Qiao, Y.; et al. A NASICON-typed $\text{Na}_4\text{Mn}_{0.5}\text{Fe}_{0.5}\text{Al}(\text{PO}_4)_3$ Cathode for Low-cost and High-energy Sodium-ion Batteries. *Carbon Neutralization* **2022**, *1*, 49–58. [[CrossRef](#)]
4. Lin, C.; Zhang, J.; von Lim, Y.; Yan, D.; Li, K.; Leong, Z.Y.; Vafakhah, S.; Li, Y.; Silva, A.; Yang, H.Y. 3D Hierarchical Architectures of CoSe_2 Nanoparticles Embedded in Rice-derived Hard Carbon for Advanced Sodium Storage. *Carbon Neutralization* **2022**, *1*, 224–232. [[CrossRef](#)]
5. Liu, X.; Zhang, L.; Yu, H.; Wang, J.; Li, J.; Yang, K.; Zhao, Y.; Wang, H.; Wu, B.; Brandon, N.P.; et al. Bridging Multiscale Characterization Technologies and Digital Modeling to Evaluate Lithium Battery Full Lifecycle. *Adv. Energy Mater.* **2022**, *12*, 2200889. [[CrossRef](#)]
6. Zhang, L.; Liu, L.; Gao, X.; Pan, Y.; Liu, X.; Feng, X. Modeling of Lithium Plating in Lithium Ion Batteries Based on Monte Carlo Method. *J. Power Source* **2022**, *541*, 231568. [[CrossRef](#)]
7. Meng, X.; Xu, Y.; Cao, H.; Lin, X.; Ning, P.; Zhang, Y.; Garcia, Y.G.; Sun, Z. Internal Failure of Anode Materials for Lithium Batteries—A Critical Review. *Green Energy Environ.* **2020**, *5*, 22–36. [[CrossRef](#)]

8. Gao, X.L.; Liu, X.H.; Xie, W.L.; Zhang, L.S.; Yang, S.C. Multiscale Observation of Li Plating for Lithium-Ion Batteries. *Rare Met.* **2021**, *40*, 3038–3048. [[CrossRef](#)]
9. Yang, Y. Production of Lithium Metal with Ion-Selective Solid Electrolytes. *Green Energy Environ.* **2020**, *5*, 382–384. [[CrossRef](#)]
10. You, H.; Zhu, J.; Wang, X.; Jiang, B.; Sun, H.; Liu, X.; Wei, X.; Han, G.; Ding, S.; Yu, H.; et al. Nonlinear Health Evaluation for Lithium-Ion Battery within Full-Lifespan. *J. Energy Chem.* **2022**, *72*, 333–341. [[CrossRef](#)]
11. Liu, X.; Ai, W.; Naylor Marlow, M.; Patel, Y.; Wu, B. The Effect of Cell-to-Cell Variations and Thermal Gradients on the Performance and Degradation of Lithium-Ion Battery Packs. *Appl. Energy* **2019**, *248*, 489–499. [[CrossRef](#)]
12. Zhou, C.-C.; Su, Z.; Gao, X.-L.; Cao, R.; Yang, S.-C.; Liu, X.-H. Ultra-High-Energy Lithium-Ion Batteries Enabled by Aligned Structured Thick Electrode Design. *Rare Met.* **2022**, *41*, 14–20. [[CrossRef](#)]
13. Pang, M.C.; Yang, K.; Brugge, R.; Zhang, T.; Liu, X.; Pan, F.; Yang, S.; Aguadero, A.; Wu, B.; Marinescu, M.; et al. Interactions Are Important: Linking Multi-Physics Mechanisms to the Performance and Degradation of Solid-State Batteries. *Mater. Today* **2021**, *49*, 145–183. [[CrossRef](#)]
14. Zhang, S.; Guo, X.; Dou, X.; Zhang, X. A Data-Driven Coulomb Counting Method for State of Charge Calibration and Estimation of Lithium-Ion Battery. *Sustain. Energy Technol. Assess.* **2020**, *40*, 100752. [[CrossRef](#)]
15. 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](#)]
16. Campestrini, C.; Kosch, S.; Jossen, A. Influence of Change in Open Circuit Voltage on the State of Charge Estimation with an Extended Kalman Filter. *J. Energy Storage* **2017**, *12*, 149–156. [[CrossRef](#)]
17. Xing, Y.; He, W.; Pecht, M.; Tsui, K.L. State of Charge Estimation of Lithium-Ion Batteries Using the Open-Circuit Voltage at Various Ambient Temperatures. *Appl. Energy* **2014**, *113*, 106–115. [[CrossRef](#)]
18. Yang, F.; Xing, Y.; Wang, D.; Tsui, K.L. A Comparative Study of Three Model-Based Algorithms for Estimating State-of-Charge of Lithium-Ion Batteries under a New Combined Dynamic Loading Profile. *Appl. Energy* **2016**, *164*, 387–399. [[CrossRef](#)]
19. Plett, G.L. Extended Kalman Filtering for Battery Management Systems of LiPB-Based HEV Battery Packs—Part 1. Background. *J. Power Source* **2004**, *134*, 252–261. [[CrossRef](#)]
20. Plett, G.L. Extended Kalman Filtering for Battery Management Systems of LiPB-Based HEV Battery Packs—Part 2. Modeling and Identification. *J. Power Source* **2004**, *134*, 262–276. [[CrossRef](#)]
21. Plett, G.L. Extended Kalman Filtering for Battery Management Systems of LiPB-Based HEV Battery Packs—Part 3. State and Parameter Estimation. *J. Power Source* **2004**, *134*, 277–292. [[CrossRef](#)]
22. Peng, J.; Luo, J.; He, H.; Lu, B. An Improved State of Charge Estimation Method Based on Cubature Kalman Filter for Lithium-Ion Batteries. *Appl. Energy* **2019**, *253*, 113520. [[CrossRef](#)]
23. Ling, L.; Sun, D.; Yu, X.; Huang, R. State of Charge Estimation of Lithium-Ion Batteries Based on the Probabilistic Fusion of Two Kinds of Cubature Kalman Filters. *J. Energy Storage* **2021**, *43*, 103070. [[CrossRef](#)]
24. Wang, Y.; He, H.; Wang, C.; Lu, Y.; Dong, K.; Huo, F.; Zhang, S. Insights into Ionic Liquids: From Z-Bonds to Quasi-Liquids. *JACS Au* **2022**, *2*, 543–561. [[CrossRef](#)] [[PubMed](#)]
25. Chaoui, H.; Ibe-Ekeocha, C.C. State of Charge and State of Health Estimation for Lithium Batteries Using Recurrent Neural Networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 8773–8783. [[CrossRef](#)]
26. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Ahmed, R.; Emadi, A. Long Short-Term Memory Networks for Accurate State-of-Charge Estimation of Li-Ion Batteries. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6730–6739. [[CrossRef](#)]
27. Hannan, M.A.; How, D.N.T.; Hossain Lipu, M.S.; Ker, P.J.; Dong, Z.Y.; Mansur, M.; Blaabjerg, F. SOC Estimation of Li-Ion Batteries with Learning Rate-Optimized Deep Fully Convolutional Network. *IEEE Trans. Power Electron.* **2021**, *36*, 7349–7353. [[CrossRef](#)]
28. 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](#)]
29. Feng, F.; Teng, S.; Liu, K.; Xie, J.; Xie, Y.; Liu, B.; Li, K. Co-Estimation of Lithium-Ion Battery State of Charge and State of Temperature Based on a Hybrid Electrochemical-Thermal-Neural-Network Model. *J. Power Sources* **2020**, *455*, 227935. [[CrossRef](#)]
30. Ragone, M.; Yurkiv, V.; Ramasubramanian, A.; Kashir, B.; Mashayek, F. Data Driven Estimation of Electric Vehicle Battery State-of-Charge Informed by Automotive Simulations and Multi-Physics Modeling. *J. Power Sources* **2021**, *483*, 229108. [[CrossRef](#)]
31. Lecun, Y.; Bottou, E.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
32. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
33. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng. Mar.* **1960**, *82*, 35–45. [[CrossRef](#)]
34. Julier, S.J.; Uhlmann, J.K. Unscented Filtering and Nonlinear Estimation. *Proc. IEEE* **2004**, *92*, 401–422. [[CrossRef](#)]
35. Zhang, L.S.; Gao, X.L.; Liu, X.H.; Zhang, Z.J.; Cao, R.; Cheng, H.C.; Wang, M.Y.; Yan, X.Y.; Yang, S.C. CHAIN: Unlocking Informatics-Aided Design of Li Metal Anode from Materials to Applications. *Rare Met.* **2022**, *41*, 1477–1489. [[CrossRef](#)]
36. Yang, S.; He, R.; Zhang, Z.; Cao, Y.; Gao, X.; Liu, X. CHAIN: Cyber Hierarchy and Interactional Network Enabling Digital Solution for Battery Full-Lifespan Management. *Matter* **2020**, *3*, 27–41. [[CrossRef](#)]
37. Yang, S.; Zhang, Z.; Cao, R.; Wang, M.; Cheng, H.; Zhang, L.; Jiang, Y.; Li, Y.; Chen, B.; Ling, H.; et al. Implementation for a Cloud Battery Management System Based on the CHAIN Framework. *Energy AI* **2021**, *5*, 100088. [[CrossRef](#)]
38. *USABC Electric Vehicle Battery Test Procedures Manual, Revision 2*; USDOE: Washington, DC, USA, 1996.

39. Pecht, M. Battery Data Set. CALCE, CALCE Battery Research Group, Maryland, MD. 2017. Available online: <https://web.calce.umd.edu/batteries/data.htm> (accessed on 12 June 2022).
40. Yang, L.; Shami, A. On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *Neurocomputing* **2020**, *415*, 295–316. [[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.