Journal of Energy Storage 70 (2023) 108037

Contents lists available at ScienceDirect

Journal of Energy Storage

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

Research papers

State of charge estimation for lithium-ion batteries based on cross-domain transfer learning with feedback mechanism

Yongsong Yang [a], Lijun Zhao [a], Quanqing Yu [a,*], Shizhuo Liu [a], Guanghui Zhou [a], Weixiang Shen [b]

[a] School of Automotive Engineering, Harbin Institute of Technology, Weihai, Shandong 264209, China
[b] School of Science, Computing and Engineering Technologies, Swinburne University of Technology, Hawthorn, Victoria 3122, Australia

ARTICLE INFO

Keywords:
Li-ion battery
State of charge
Transfer learning
Deep learning
Error feedback mechanism

ABSTRACT

When the deep learning model is applied to estimate battery state of charge (SOC), the information inside the training set cannot be leveraged thoroughly, which would cause poor SOC estimation accuracy and robustness on the testing set. To solve the problem, this paper proposes an adaptive convolutional neural network-gated recurrent unit with Kalman filter and feedback mechanism (Fb-Ada-CNN-GRU-KF) for SOC estimation considering distribution difference of data segments inside the training set through transfer learning and extracting the spatial information through convolutional layer. Furthermore, the feedback mechanism provides the model more information to learn to correct the systematic error, and the KF in the proposed model works as a post data processor to obtain a steady and smooth SOC estimation results. Experimental and comparison results show that the proposed model for SOC estimation outperforms the existing deep learning methods in terms of the accuracy, generalization and stability.

1. Introduction

Globally, fuel combustion in transportation produces 24 % of carbon emissions, which is a major cause of global warming [1]. Electric vehicles (EVs) have been providing a solution to reduce emissions and mitigate climate change [2]. To ensure the safety and reliability of lithium ion (Li-ion) batteries in EVs, SOC, which signifies the remaining capacity of a Li-ion battery [3,4], needs to be estimated accurately in battery management systems (BMSs) [5]. Due to the strong nonlinear characteristics of Li-ion batteries, accurate SOC estimation is still facing significant challenges [6,7].

To date, the most popular SOC estimation method includes coulomb counting method, open circuit voltage (OCV) method, equivalent circuit model (ECM) method, and data-driven method. Coulomb counting method is restricted by high quality sensors and the unknown initial SOC [8]. OCV method can only be used when batteries reach equilibrium condition, which takes a long time and has a poor accuracy in the middle range of SOC [9]. ECM method usually combined with Kalman Filter (KF) requires to make the balance between model complexity and estimation performance [10–12]. Therefore, researchers attempt to explore data-driven method for a potential solution [13].

The most basic and popular model of data-driven method includes backpropagation neural network (BPNN), recurrent neural network (RNN) and the combination of RNN and convolutional neural network (CNN), etc. [14]. He et al. [15] introduced the BPNN algorithm to estimate the SOC of a lithium iron phosphate battery, and the root mean square error (RMSE) of the predicted SOC was reported at 3.3 % under US06 driving cycle at 25 °C. Ephrem Chemali et al. [16] used long short term memory (LSTM) neural network, one kind of RNN, for SOC estimation. The application of LSTM overcame the drawback of exploding or vanishing gradient during back-propagation training of traditional RNN. Huang et al. [17] proposed a SOC estimation method using the CNN-GRU model, consisting of CNN layer followed by gated recurrent unit (GRU) layer and fully connected (FC) layer, achieving a mean absolute error (MAE) of 1.68 % under US06 diving cycle at 30 °C. The combination of CNN and GRU helps in extracting both spatial features and temporal features in data. Liu et al. [18] developed a SOC estimation method leveraging transfer learning with fine-tuning strategy, which needs only 30 % of training data when transferred to different batteries. Oyewole et al. [19] utilized a controllable deep transfer learning (CDTL)

[*] Corresponding author at: School of Automotive Engineering, Harbin Institute of Technology, Weihai, No. 2, West Wenhua Road, High-tech District, Weihai 264209, Shandong, China.
E-mail address: qqyu@hit.edu.cn (Q. Yu).

https://doi.org/10.1016/j.est.2023.108037
Received 4 November 2022; Received in revised form 14 April 2023; Accepted 10 June 2023
Available online 21 June 2023

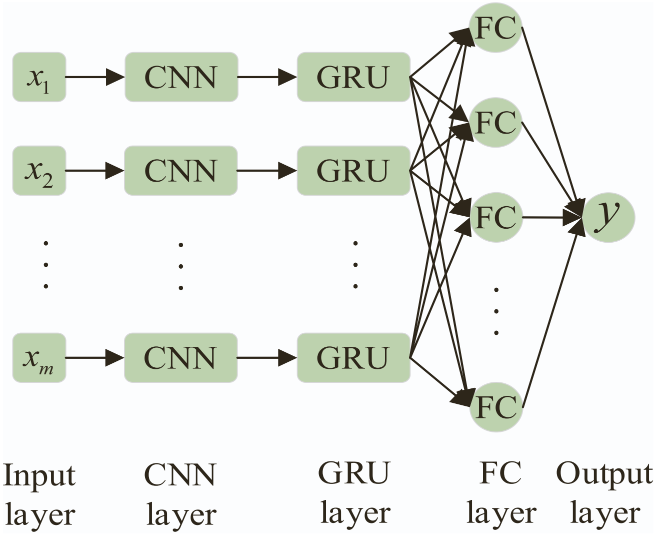

Fig. 1. Overall structure of CNN-GRU model.

network on the shared knowledge between a target battery and a source battery with adaptive regularization, which guarantees the controllability and reduces the likelihood of negative transfer learning. According to the results, the CDTL method has an improvement of 60 % in RMSE for the battery with the same chemistry.

However, none of the above-mentioned methods has thoroughly considered the distribution differences in the training dataset, which causes a poor performance in generalization and accuracy in SOC prediction results. In the field of artificial intelligence theory, Du et al. [20] recently proposes an Ada-RNN method, which incorporate RNN and transfer learning to characterize the distribution difference in time series which leverages the principle of maximum entropy. The results show that the Ada-RNN has strong ability to provide the generalized and accurate SOC estimation. Consequently, an effective SOC estimation framework called the feedback Fb-Ada-CNN-GRU-KF method, based on deep learning model with transfer learning, feedback mechanism and post data processor, is proposed in this paper. The contributions of this paper can be summarized as follows:

(1). Ada-CNN-GRU model based on Ada-RNN with transfer learning function is proposed and firstly applied for SOC estimation of a Li-ion battery, which takes into consideration of both the segments' distribution difference and spatial features of the training dataset.

(2). An error feedback mechanism, taking the predicted SOC error as an additional training feature, is proposed to enrich the information of training dataset and to reduce systematic prediction error. The KF algorithm is leveraged as a post processing machine to eliminate the abnormal prediction value and suppress its fluctuation.

(3). The accuracy and robustness of the proposed method for SOC estimation is verified and compared with that of the other existing methods under the dynamic driving cycle from Panasonic Li-ion battery at different temperatures. Also, the effectiveness of adding CNN layer, KF algorithm and feedback mechanism into the proposed model to improve the performance of SOC estimation is verified.

The remaining sections of this paper are organized as follows: Section 2 introduces the methodology: the architecture of the CNN-GRU; the architecture of the Ada-GRU; and the KF algorithm. Section 3 elaborates the Fb-Ada-CNN-GRU-KF method for SOC estimation. Section 4 describes the experimental analysis: dataset description; input and output structures; and evaluation results at different ambient temperatures. Section 5 is the conclusion of this study.

2. Methodology

2.1. The architecture of the CNN-GRU

The overall structure of CNN-GRU model is shown in Fig. 1, including an input layer, a CNN layer, a GRU layer, a fully connected (FC) layer and an output layer. It is mainly a combination of CNN model and GRU model.

CNN is a class of deep learning neural network with the advantages of local connection, weight sharing and down-sampling dimensionality reduction [21] can extract features adaptively [1] and reduce the computation burden. The core of the CNN is the convolution kernel, through which the convolution is operated. As shown in Fig. 2(a), CNN extracts feature information through moving convolution kernel. Assuming that the input and output of the convolutional neural network are $x_{d,i+m,j+n}$ and $a_{i,j}$, respectively, the relationship between the input and output can be expressed as follows:

$$a_{i,j} = f\left(\sum_{d=0}^{D-1}\sum_{w=0}^{W-1}\sum_{h=0}^{H-1} w_{d,w,h}x_{d,i+m,j+n} + w_b\right) \tag{1}$$

where D, W and H respectively denotes the output depth, width and height of convolutional kernal, $f\,(\cdot)$is the convolution operation.

As an improved RNN, GRU can avoid the problem of gradient disappearance [22] and allow each recurrent unit to adaptively capture sequence dependencies over various lengths of time [17]. Fig. 2(b) shows a GRU unit which contains the reset gate, which determines how much past information will be forgotten; and the update gate, which determines how much information will be updated. At the time step t, the forward pass of a GRU unit is proceeded as follows:

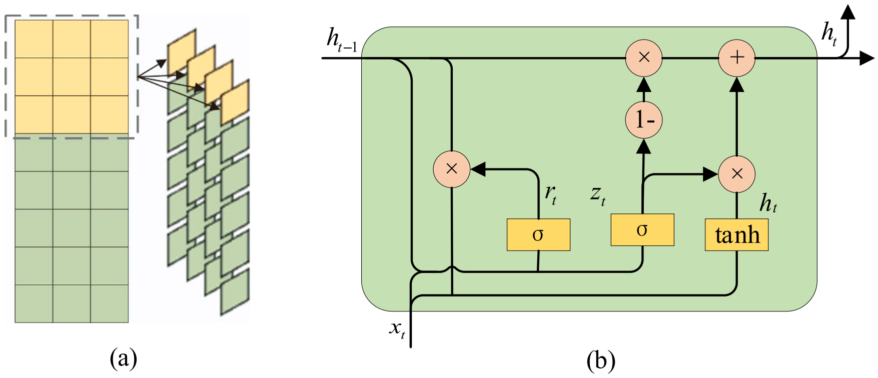

Fig. 2. Structure of (a) CNN (b) GRU.

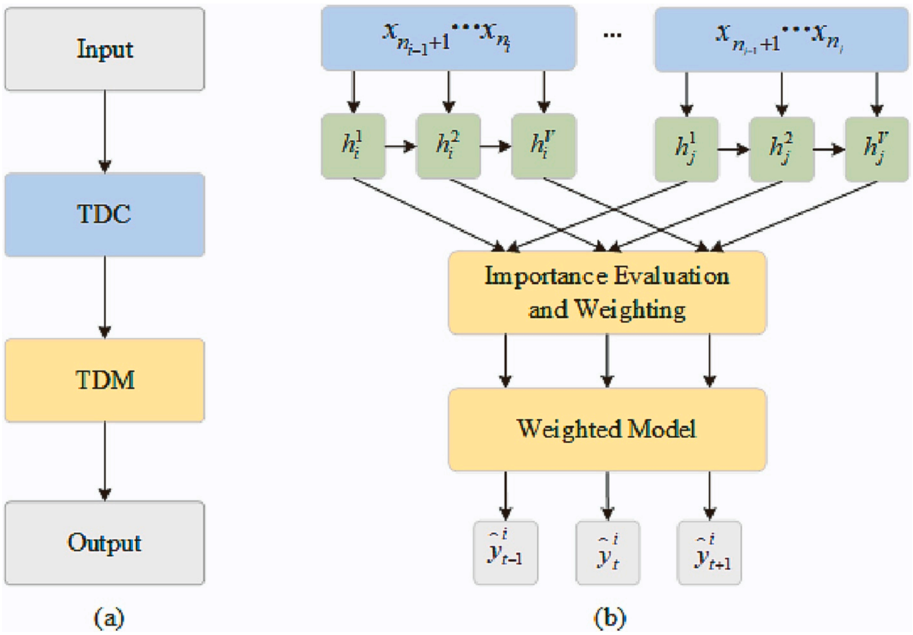

Fig. 3. The architecture of (a) Ada-RNN (b) TDM.

$$\begin{cases} r_t = \sigma(w_r \cdot [h_{t-1}, x_t]) \\ z_t = \sigma(w_z \cdot [h_{t-1}, x_t]) \\ \widetilde{h}_t = tanh(w \cdot [r_t \odot h_{t-1}, x_t]) \\ h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t \end{cases} \tag{2}$$

where r_t and z_t are the reset gate and update gate, respectively; x_t is the input; $w$represents the weight value; $\widetilde{h}_t$ and h_t are the temporary hidden state and output of the hidden layer, respectively. $\sigma(\cdot)$ and $tanh(\cdot)$ are the sigmoid activation function and bitangent activation function, respectively; $\odot$ represents an element-wise multiplication.

2.2. The architecture of the Ada-RNN

Ada-RNN proposed by Du et al. [20] is a class of transfer learning method which can find the data segments with the largest distribution difference in the training dataset and reduce the distribution difference between these data segments to achieve higher generalization and accuracy than the traditional RNN model. The overview of Ada-RNN is depicted in Fig. 3(a), it is mainly composed of two parts: temporal distribution characterization (TDC) and temporal distribution matching (TDM).

The TDC algorithm aims at finding the data segments with the largest distribution differences and splitting them apart based on the principle of maximum entropy, and it searches for the periods most dissimilar to each other through the following formula:

$$\begin{aligned} &\underset{0<K\leq K_0}{max}\ \underset{n_1,\dots,n_K}{max} \frac{1}{K} \sum_{1\leq i\neq j\leq K} d(D_i, D_j) \\ &st.\forall i, \Delta_1 < |D_i| \langle \Delta_2; \sum_i |D_i| = n \end{aligned} \tag{3}$$

where K is the number of periods, and K_0 is a hyper-parameter to avoid over-splitting; $d(\cdot,\cdot)$ is a distance metric; D is the time series segments, Δ_1 and Δ_2 are hyper-parameters to avoid trivial solutions.

In addition, $d(\cdot,\cdot)$ can be any distance function, such as maximum mean discrepancy (MMD) [23] or cosine distance [24]. Taking the MMD function as an example, the distance is calculated as follows:

$$d_{mmd}(x_s, x_t) = \frac{1}{n_s^2} \sum_{i,j=1}^{n_s} k(x_{s_i}, x_{s_j}) + \frac{1}{n_t^2} \sum_{i,j=1}^{n_t} k(x_{t_i}, x_{t_j}) - \frac{2}{n_s n_t} \sum_{i,j=1}^{n_s+n_t} k(x_{s_i}, x_{t_j}) \tag{4}$$

where x_s and x_t are the source data and target data, respectively; $k(\cdot,\cdot)$ is the radial basis function (RBF) kernel; n_s and n_t are the number of the two sets of data.

To avoid complicated calculation, the TDC algorithm will evenly divide the time series into N parts, in which each part is the smallest unit and undividable. Then, an appropriate value of K, which is an integer, will be determined in a range of [2, N] through greedy algorithm. For example, if we set the starting point of the data to be A, the end point to be B and N to be 5. When $K = 2$, our goal is to find an appropriate point C to maximize the distance $d(S_{AC}, S_{CB})$. The TDC algorithm is generally applied to the case which has no obvious distribution differences in the time series. If a period of time series itself has obvious distribution difference limits, different data segments can be divided manually, and the goal of getting data segments with the largest distribution difference will also be achieved.

TDM in the Ada-RNN model is responsible for matching these split data segments and building a prediction model, and it is implemented by minimizing the followng loss function:

$$L(\theta, \alpha) = L_{pred}(\theta) + \lambda \frac{2}{K(K-1)} \sum_{i,j}^{i\neq j} L_{tdm}(D_i, D_j; \theta, \alpha) \tag{5}$$

where θ denotes the learnable model parameters; α denotes the importance evaluation factor of each pair of data segments; and λ is a trade-off parameter.

Specially, the $L_{pred}(\theta)$ can be computed by

$$L_{pred}(\theta) = \frac{1}{K} \sum_{j=1}^{K} \frac{1}{|D_j|} \sum_{i=1}^{|D_j|} l(y_i^j, M(x_i^j; \theta)) \tag{6}$$

where $l(\cdot,\cdot)$ is a MSE loss function; M denotes the prediction model; and (x_i^j, y_i^j) denotes the i-th labeled segment from period D_j.

Given a period of (D_i, D_j), loss of temporal distribution matching can be obtained by

$$L_{tdm}(D_i, D_j; \theta, \alpha_{i,j}^t) = \sum_{t=1}^{V} \alpha_{i,j}^t d(h_i^t, h_j^t; \theta) \tag{7}$$

where (h_i^t, h_j^t) denotes the hidden state parameters of models trained

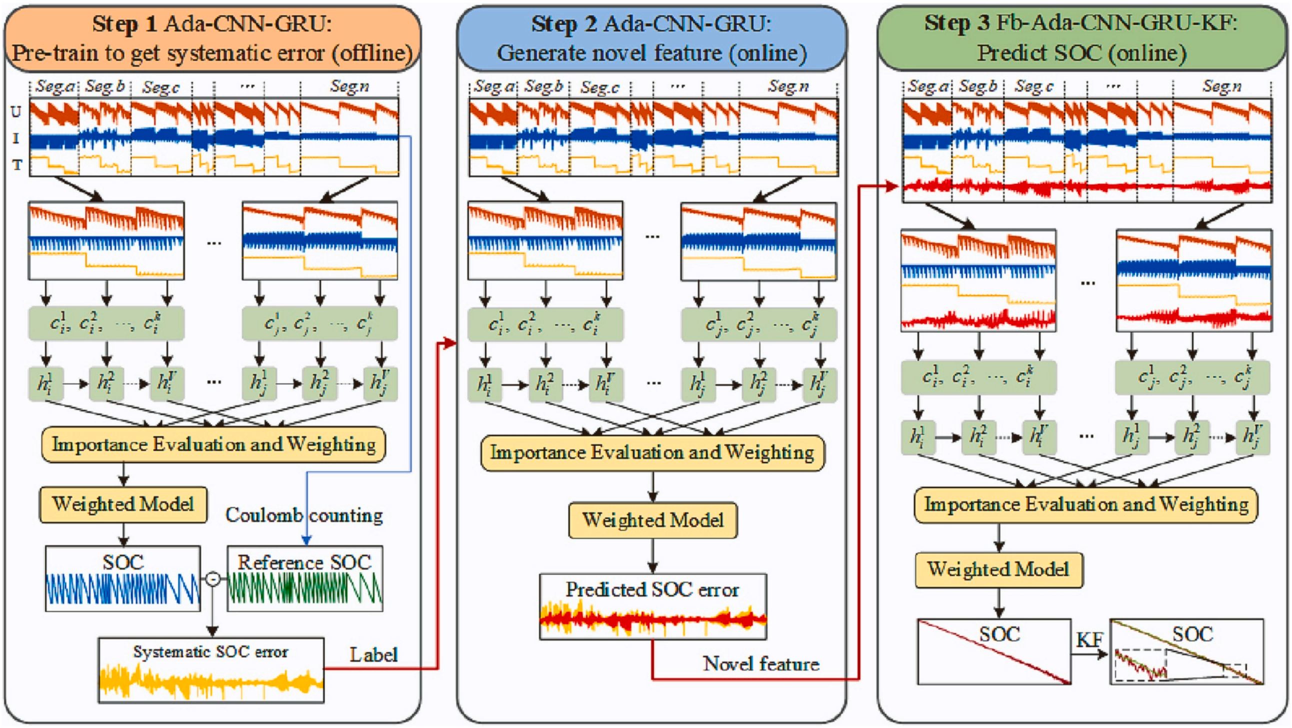

Fig. 4. The proposed framework of Fb-Ada-CNN-GRU-KF model.

from D_i andD_j, respectively.

The evaluation factor of each pair of data segments, $\alpha_{i,j}^{t}$, aiming at learning the relative importance of V-dimensional hidden states in the model, is calculated by

$$\alpha_{i,j}^{t,(n+1)} \begin{cases} \alpha_{i,j}^{t,(n)} \times \left(1+\sigma\left(d_{i,j}^{t,(n)} \times d_{i,j}^{t,(n-1)}\right)\right) & d_{i,j}^{t,(n)} \geq d_{i,j}^{t,(n-1)} \\ \alpha_{i,j}^{t,(n)} & \text{otherwise} \end{cases} \tag{8}$$

where $d_{i,j}^{t,(n)}$ is the distribution distance at the time step t in epoch n.

2.3. *The KF algorithm*

The KF algorithm obtains the optimal estimation of the current moment using the estimated value of the previous moment and the observed value of the current moment. It is composed of two parts: time update equation and state update equation. The time update equation, shown in Eq. (9), is used to extrapolate forward the values of the current state variables and error covariance estimates in time in order to construct prior estimates for the next state [25]:

$$\begin{cases} \widehat{x}_{n+1,n} = F\widehat{x}_{n,n} + Gu_n \\ P_{n+1,n} = FP_{n,n}F^T + Q \end{cases} \tag{9}$$

where $\widehat{x}_{n+1,n}$ is a predicted system state vector at the time step $n+1$, $\widehat{x}_{n,n}$ is an estimated system state vector at the time step n, u_n represents input variable, F is the state transition matrix, G represents control matrix; $P_{n+1,n}$is the covariance matrix for the next state, $P_{n,n}$is the covariance matrix of the current state, Q is the process noise uncertainty.

The state update equation incorporates a new measurement into the a priori estimate to obtain an improved posteriori estimate [25], and it is computed as:

$$\begin{cases} K_n = P_{n,n-1}H^T\left(HP_{n,n-1}H^T + R_n\right)^{-1} \\ \widehat{x}_{n,n} = \widehat{x}_{n,n-1} + K_n\left(z_n - H\widehat{x}_{n,n-1}\right) \\ P_{n,n} = (I - K_nH)P_{n,n-1}(I - K_nH)^T + K_nR_nK_n^T \end{cases} \tag{10}$$

where K_n is Kalman gain; H is observation matrix, R_n is the measurement uncertainty; z_n is a measurement; $I$is an identity matrix.

3. Fb-Ada-CNN-GRU-KF model for SOC estimation

RNN module has a problem of gradient disappearance in long time series prediction, to solve this problem, the RNN module in Ada-RNN would be substituted with GRU in this paper. And for a better spatial information extraction, a CNN layer would be added before the GRU layer, so that an Ada-CNN-GRU model is built.

For a certain deep learning model, the distribution of training data and the structure of the model itself can lead to a certain error in different validation set after training, namely systematic error, which means it can be predicted theoretically. Therefore, an error feedback mechanism is proposed to first predict the systematic error and feed it back as input for model correction.

As too many parameters in neural network can result in overfitting and output fluctuation in complex prediction problems, and Kalman filter can effectively decrease fluctuations considering both the observed value and the predicted value, thus is adopted as the post-processor to reduce the fluctuation of model prediction.

The proposed framework of the Fb-Ada-CNN-GRU-KF model for SOC estimation is shown in Fig. 4, which is mainly composed of 3 steps.

Step 1: an Ada-CNN-GRU model will be roughly trained to obtain the predicted SOC, then the SOC error on the training data set, which partly reflects the model's systematic accuracy, would easily be calculated according to reference SOC.

Step 2: an Ada-CNN-GRU model, which takes the SOC error obtained from the step one as label, will be trained to make a prediction of SOC error on the validation and testing data set. Thus, the SOC error can be used as a novel feature for the feedback on these two data sets.

Step 3: a Fb-Ada-CNN-GRU model for SOC estimation will be trained, which takes the voltage, current, temperature of a battery and the predicted SOC error as input and the SOC as an output. After that, the KF is added at the end of the proposed framework as a post processor, which takes the SOC from the Fb-Ada-CNN-GRU model as measurement

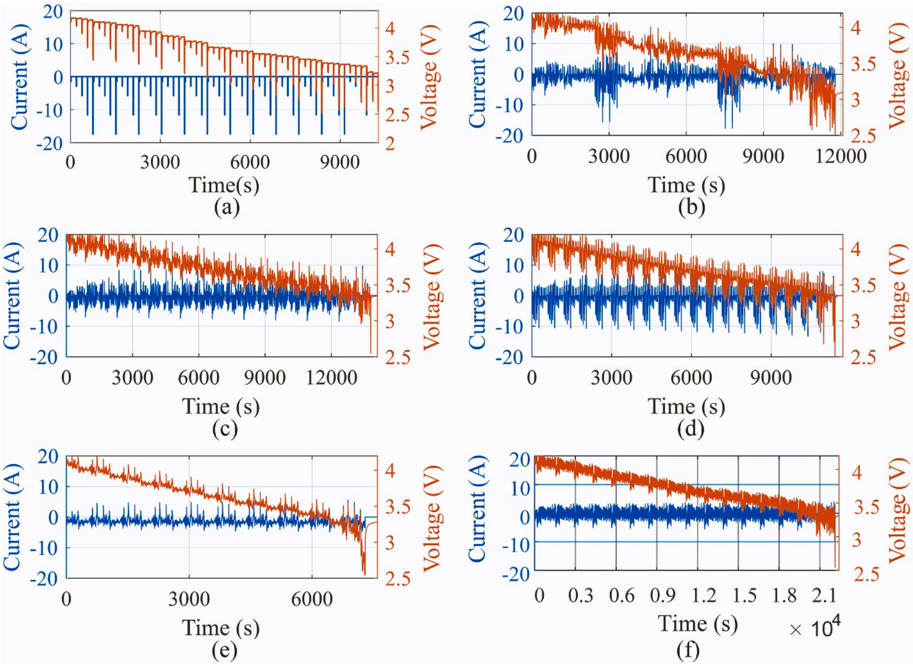

Fig. 5. Current and voltage profiles based on the dynamic driving cycles at 25 °C (a) HPPC (b) Cycle4 (c) LA92 (d) NN (e) HWFET (f) UDDS.

value and the Coulomb Counting method as the update function to further improve the stability of the SOC estimation.

The maximum absolute error (MAX), mean absolute error (MAE) and the root mean square error (RMSE), as defined in Eq. (11), are chosen to assess the SOC estimation performance of the proposed framework. The MMD, as shown in Eq. (4), is chosen to be the distance calculation formula between each pair of segments.

$$\begin{cases} MAX = max(|y_i - \widehat{y}_i|) \\ \mathrm{MAE} = \dfrac{1}{m}\sum_{i=1}^{m}|y_i - \widehat{y}_i| \\ \mathrm{RMSE} = \sqrt{\dfrac{1}{m}\sum_{i=1}^{m}(y_i - \widehat{y}_i)^2} \end{cases} \tag{11}$$

4. Experimental analysis

4.1. Dataset description

The data used in this paper is recorded by Phillip [26] from a Panasonic NCR18650PF Li-ion battery, of which the nominal capacity is 2.32 Ah. The battery was tested under the selected drive cycles at different ambient temperatures, and the whole test was conducted in a thermal chamber with testing equipment manufactured by Digatron Firing Circuits. The current sensor used to measure current and calculate capacity has an error of less than 25 mA. For the typical dataset, this can cause a cumulative error under 40 mAh, which is very small compared to the battery capacity [16].

The thermal chamber was first set to 25 °C followed by 3 h of rest to guarantee the battery's internal temperature is 25 °C. The battery was then fully charged with a constant current of 1C followed by a constant voltage of 4.2 V. This charge process was terminated when the charge current fell below 50 mA. The thermal chamber was then set to the desired temperature and waited for another 3 h for the battery's internal temperature to stabilize. After that, the dynamic driving cycle test would start, including HWFET, HPPC, LA92, US06, UDDS and the mixture of them named as Cycle4 and NN. Fig. 5 shows the current and voltage profiles based on these dynamic driving cycles which differ greatly from each other and will provide a broad range of realistic driving conditions for model training and validation.

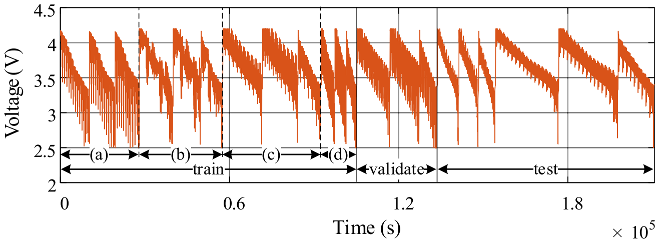

Fig. 6. Dataset division and TDC operation on battery data.

4.2. Evaluation at different ambient temperatures

The battery data is divided into the training dataset, validation dataset and testing dataset. The training dataset for SOC estimation consists of the data corresponding to 4 dynamic driving cycles: HPPC, Cycle 4, LA92, US 06, the validation dataset consists of the data corresponding to NN cycle, and the testing dataset includes the data recorded in HWFET cycle and UDDS cycle. Since the characteristic of the battery is heavily affected by temperature, all of the data contains the information of the battery at the temperatures of 25 °C, 10 °C and 0 °C. As the data used in this paper are from a laboratory, it is easy to identify the distribution difference in all the dataset adopting the manual method of TDC to split the time series and the results are shown in Fig. 6. And letter a, b c, d in Fig. 6 represent driving conditions of HPPC, Cycle 4, LA 92 and US06 under 3 different temperatures, respectively.

For a long input sequence length can result in a better estimation accuracy within limits but a huge computing consumption and lower computing speed, a balance is needed when setting the length of input sequence. According to experience, the input is set as a matrix with a fixed size of 120 × 3 and the corresponding output is a 1 × 1 SOC value, they can be expressed as:

Table 1
Structure hyper-parameters of the proposed model.

Layers (in order)	Parameter Information	Settings
CNN	(Input channel, output channel, kernel size)	(1, 4, (5, 3))
CNN	(Input channel, output channel, kernel size)	(4, 8, 5)
GRU	(Input size, hidden size, layer number)	(8, 32, 2)
FC	(Input size, output size)	(32,1)

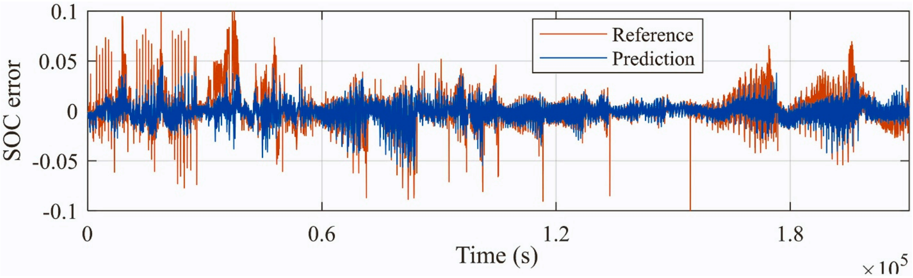

Fig. 7. Prediction of SOC error.

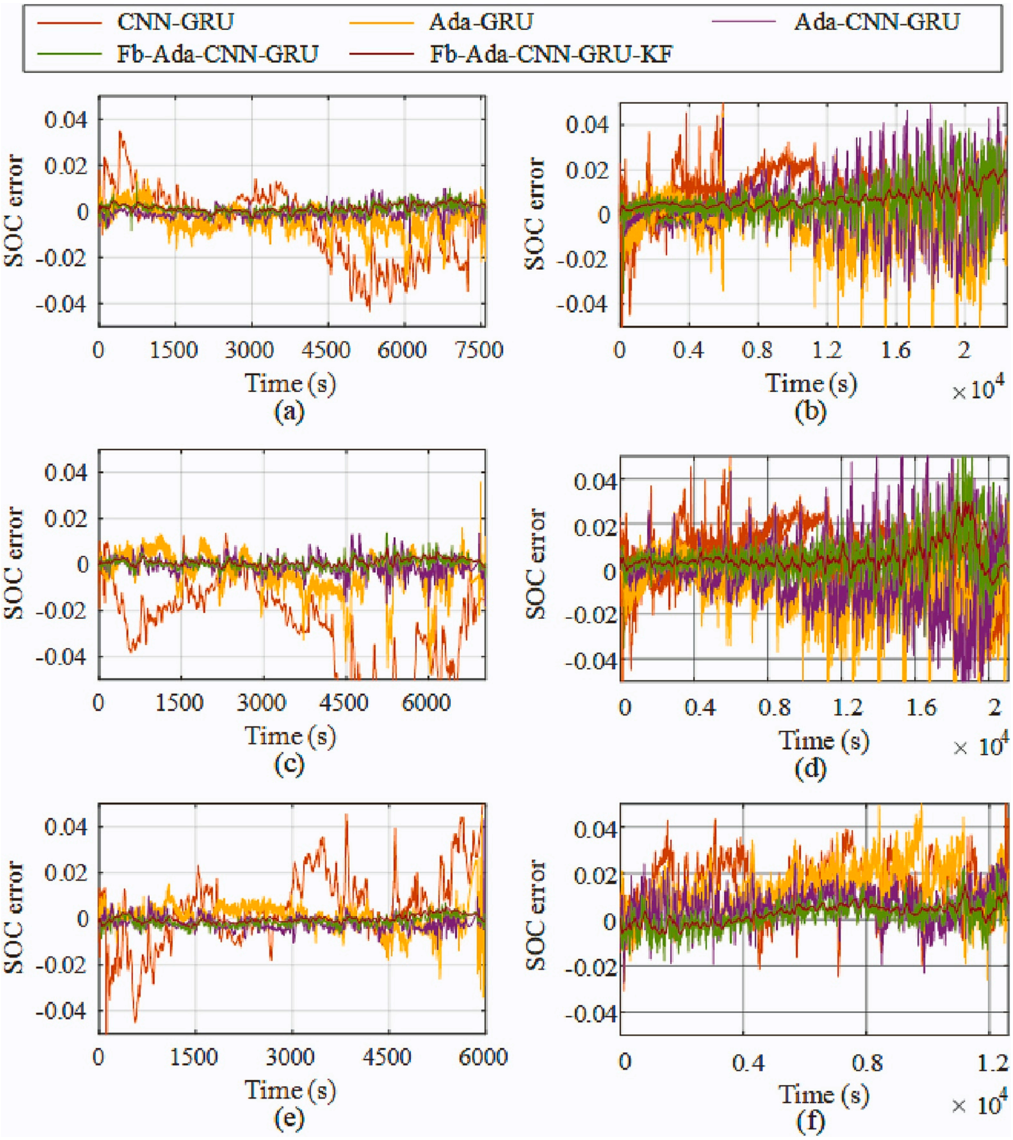

Fig. 8. Evaluation results of HWFET cycle at (a) 25 °C, (c) 10 °C, (e) 0 °C, of UDDS cycle at (b) 25 °C, (d) 10 °C, (f) 0 °C.

Table 2
Evaluation results under HWFET cycle.

	25 °C			10 °C			0 °C		
Models	MAE (%)	RMSE (%)	MAX (%)	MAE (%)	RMSE (%)	MAX (%)	MAE (%)	RMSE (%)	MAX (%)
#1	1.29	1.67	4.34	2.80	3.41	7.52	1.31	1.72	6.55
#2	0.59	2.24	3.06	0.79	2.81	4.75	0.50	2.23	4.52
#3	0.22	1.16	1.38	0.30	1.77	1.81	0.40	3.47	4.33
#4	0.19	0.24	1.29	0.17	0.23	1.38	0.19	0.23	0.86
#5	0.18	0.22	0.60	0.16	0.20	0.73	0.13	0.17	0.54

Note: #1: CNN-GRU, #2: Ada-GRU, #3 Ada-CNN-GRU, #4: Fb-Ada-CNN-GRU, #5: Fb-Ada-CNN-GRU-KF. The same model number is also used in the following table.

Table 3
Evaluation results under UDDS cycle.

	25 °C			10 °C			0 °C		
Models	MAE (%)	RMSE (%)	MAX (%)	MAE (%)	RMSE (%)	MAX (%)	MAE (%)	RMSE (%)	MAX (%)
#1	1.22	1.39	3.69	1.00	1.28	4.66	1.64	1.88	5.22
#2	0.86	1.62	6.59	1.29	2.16	7.71	1.47	1.73	7.59
#3	0.98	1.48	4.94	1.25	1.93	7.05	0.71	0.87	2.72
#4	0.74	1.04	4.20	0.62	0.93	5.26	0.42	0.53	2.32
#5	0.70	0.82	2.12	0.54	0.78	2.98	0.39	0.47	1.52

$$\begin{cases} input_i = \begin{pmatrix} U_i & I_i & T_i \\ U_{i+1} & I_{i+1} & T_{i+1} \\ \vdots & \vdots & \vdots \\ U_{i+119} & I_{i+119} & T_{i+119} \end{pmatrix}_{120\times 3} \\ output_i = (SOC_i)_{1\times 1} \end{cases} \tag{12}$$

where U_i, I_i, T_i, SOC_i denote the voltage, current, temperature and SOC at the i-th sampling point, respectively, 120 rows indicate that the input window contains 120 sampling point, and 3 columns indicate the feature number is 3.

In order to investigate the effect and superiority of the proposed Fb-Ada-CNN-GRU-KF model for SOC estimation, we compare it with the CNN-GRU model and Ada-GRU model for SOC estimation. We also investigate the effect of every single part of the modification of the Fb-Ada-CNN-GRU-KF model. It is worth noticing that the reference SOC is calculated using the coulomb counting method.

Neural networks are sensitive to hyper-parameters, even the same model can output totally different results because of the difference of hyper-parameters. In this paper, the structure hyper-parameters such as the size and stride of convolution kernel in CNN layer, the number of layers and hidden nodes of GRU layers of the proposed method are kept as same as possible with the comparison model. And the specific settings are shown in Table 1. Note that when there is novel feature added as input in Step 3, the kernel size of the first CNN layer would be (5, 4).

Learning rate and batch size are fine-tuned to make sure that each model could bring out its best result, they are set to 0.001 and 16, respectively. And training epoch is set to 300 in every Step. In addition, the weight of transfer learning loss is set to 0.01. In the post processor KF algorithm, the process noise and measurement noise are both set to 10^{-5} according to the parameter tuning experience gained on validation set.

Fig. 7 shows the SOC error of all the training set, validation set and test set after Step 2. And it will be then used as a novel feedback feature in Fb-Ada-CNN-GRU model and Fb-Ada-CNN-GRU-KF model. It is worth noticing that the SOC error in test set was predicted based on voltage, current and temperature, and thus it can be leveraged directly under unknown conditions.

The predicted SOC error results under HWFET and UDDS driving cycles at different temperatures of all the comparative models and the proposed model are shown in Fig. 8, their statistical results are illustrated in Tables 2 and 3, respectively.

From Fig. 8, it can be observed that the CNN-GRU model always has the largest MAE, RMSE and MAX error under HWFET and UDDS driving cycle under each working condition. By contrast, the Ada-GRU model with transfer learning has a relatively small MAE and RMSE, but there is still a large fluctuation. The Ada-CNN-GRU method by adding the CNN layers before GRU layers on the Ada-GRU model can take the spatial characteristics of the data into account and achieve a MAE of 0.5 % under HWFET driving cycle and 1.5 % under UDDS driving cycle. The Fb-Ada-CNN-GRU model through introducing a feedback mechanism to the Ada-CNN-GRU model can achieve a MAE of 0.2 % under HWFET driving cycle and 0.8 % under UDDS driving cycle, which means that the accuracy is further improved. The Fb-Ada-CNN-GRU-KF model by employing the KF as a post processor can achieve a MAE below 0.2 % and MAX error below 0.8 % under HWFET driving cycle and a MAE below 0.8 %, MAX error below 3 % under UDDS driving cycle.

From the above discussion, the proposed Fb-Ada-CNN-GRU-KF model significantly improve its performances in terms of the accuracy, generalization and stability in SOC estimation of a Li-ion battery through the introduction of transfer learning, spatial feature extraction, feedback mechanism and the KF as a post processor compared to traditional CNN-GRU models.

5. Conclusion

In this paper, a Fb-Ada-CNN-GRU-KF method, which combines the advantages of transfer learning and deep learning, is proposed based on Ada-GRU model and CNN-GRU model. To further improve the accuracy, a feedback mechanism is introduced to predict the SOC error as a novel feature for model training and the KF is implemented as a post data processor to provide a steady and smooth SOC estimation results. The accuracy and robustness of the proposed model for SOC estimation are verified under HWFET and UDDS working conditions at different ambient temperatures and compared with the CNN-GRU model, Ada-GRU model, Ada-CNN-GRU model and Fb-Ada-CNN-GRU model. The results show that the proposed model has a minimum mean MAE and RMSE values of 0.78 % and 0.82 % under HWFET and UUDS driving cycle and achieve the best performances in terms of the accuracy, generalization and stability in SOC estimation of a Li-ion battery. The results also validate the effectiveness of every part of the modification based on CNN-GRU model and Ada-GRU model.

Future work could (1) extend the application of the proposed model to SOH estimation of batteries and further improve the accuracy of SOC estimation based on the estimated SOH; (2) optimize the parameters that need to be manually adjusted.

CRediT authorship contribution statement

Yongsong Yang: Methodology, Writing - Original draft preparation.
Lijun Zhao: Supervision.
Quanqing Yu: Writing - Reviewing and Editing.
Shizhuo Liu: Conceptualization,
Guanghui Zhou: Validation.
Weixiang Shen: Writing - Reviewing and Editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This work was jointly supported by the Natural Science Program of Shandong Province (Grant No. ZR2020ME209) and National Natural Science Foundation of China (Grant No. 52177210).

References

[1] M.S. Hossain Lipu, M.A. Hannan, A. Hussain, A. Ayob, M.H.M. Saad, T.F. Karim, D. N.T. How, Data-driven state of charge estimation of lithium-ion batteries: algorithms, implementation factors, limitations and future trends, J. Clean. Prod. 277 (2020), 124110.
[2] H. He, F. Sun, Z. Wang, C. Lin, C. Zhang, R. Xiong, J. Deng, X. Zhu, P. Xie, S. Zhang, Z. Wei, China's battery electric vehicles lead the world: achievements in technology system architecture and technological breakthroughs, Green Energy Intell. Transp. 1 (2022) 100020.
[3] P. Takyi-Aninakwa, S. Wang, H. Zhang, H. Li, W. Xu, C. Fernandez, An optimized relevant long short-term memory-squared gain extended Kalman filter for the state of charge estimation of lithium-ion batteries, Energy. 260 (2022), 125093.
[4] Y. Zhang, M. Zhao, Cloud-based in-situ battery life prediction and classification using machine learning, Energy Storage Mater. 57 (2023) 346–359.
[5] Z. Fang, Z. Chen, Q. Yu, B. Zhang, R. Yang, Online power management strategy for plug-in hybrid electric vehicles based on deep reinforcement learning and driving cycle reconstruction, Green Energy Intell. Transp. 1 (2022), 100016.
[6] C. Li, F. Xiao, Y. Fan, An approach to state of charge estimation of lithium-ion batteries based on recurrent neural networks with gated recurrent unit, Energies. 12 (2019) 1592.
[7] Q. Yu, Y. Liu, S. Long, X. Jin, J. Li, W. Shen, A branch current estimation and correction method for a parallel connected battery system based on dual BP neural networks, Green Energy Intell. Transp. 1 (2022), 100029.
[8] F. Mohammadi, Lithium-ion battery state-of-charge estimation based on an improved coulomb-counting algorithm and uncertainty evaluation, J. Energy Storage. 48 (2022), 104061.
[9] Q. Yu, Y. Huang, A. Tang, C. Wang, W. Shen, OCV-SOC-temperature relationship construction and state of charge estimation for a series– parallel Lithium-ion battery pack, IEEE Trans. Intell. Transp. Syst. 24 (2023) 6362–6371.
[10] Y. Shi, S. Ahmad, Q. Tong, T.M. Lim, Z. Wei, D. Ji, C.M. Eze, J. Zhao, The optimization of state of charge and state of health estimation for lithium-ions battery using combined deep learning and Kalman filter methods, Int. J. Energy Res. 45 (2021) 11206–11230.
[11] Q. Yu, L. Dai, R. Xiong, Z. Chen, X. Zhang, W. Shen, Current sensor fault diagnosis method based on an improved equivalent circuit battery model, Appl. Energy 310 (2022), 118588.
[12] H. Yang, X. Sun, Y. An, X. Zhang, T. Wei, Y. Ma, Online parameters identification and state of charge estimation for lithium-ion capacitor based on improved Cubature Kalman filter, J. Energy Storage. 24 (2019), 100810.
[13] Z. Ni, Y. Yang, A combined data-model method for state-of-charge estimation of Lithium-ion batteries, IEEE Trans. Instrum. Meas. 71 (2022) 2503611.
[14] S. Peng, H. Zhang, Y. Yang, B. Li, S. Su, S. Huang, G. Zheng, Spatialoral dynamic forecasting of EVs charging load based on DCC-2D, Chinese, J. Electr. Eng. 8 (2022) 53–62.
[15] W. He, N. Williard, C. Chen, M. Pecht, State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation, Int. J. Electr. Power Energy Syst. 62 (2014) 783–791.
[16] E. Chemali, P.J. Kollmeyer, M. Preindl, R. Ahmed, A. Emadi, Long short-term memory networks for accurate state-of-charge estimation of Li-ion batteries, IEEE Trans. Ind. Electron. 65 (2018) 6730–6739.
[17] Z. Huang, F. Yang, F. Xu, X. Song, K.-L. Tsui, Convolutional gated recurrent unit–recurrent neural network for state-of-charge estimation of lithium-ion batteries, IEEE Access. 7 (2019) 93139–93149.
[18] Y. Liu, X. Shu, H. Yu, J. Shen, Y. Zhang, Y. Liu, Z. Chen, State of charge prediction framework for lithium-ion batteries incorporating long short-term memory network and transfer learning, J. Energy Storage. 37 (2021), 102494.
[19] I. Oyewole, A. Chehade, Y. Kim, A controllable deep transfer learning network with multiple domain adaptation for battery state-of-charge estimation, Appl. Energy 312 (2022), 118726.
[20] Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, C. Wang, AdaRNN: adaptive learning and forecasting of time series, in: Proc. 30th ACM Int. Conf. Inf. Knowl. Manag., Association for Computing Machinery, New York, NY, USA, (2021) 402–411.
[21] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: analysis, applications, and prospects, IEEE Trans. Neural Networks Learn. Syst. (2021), https://doi.org/10.1109/TNNLS.2021.3084827.
[22] X. Song, F. Yang, D. Wang, K.-L. Tsui, Combined CNN-LSTM network for state-of-charge estimation of lithium-ion batteries, IEEE Access. 7 (2019) 88894–88902.
[23] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, B.K. Sriperumbudur, Optimal kernel choice for large-scale two-sample tests, in: Adv. Neural Inf. Process 2, Syst., Curran Associates, Inc, 2012, pp. 1205–1213.
[24] B. Sun, K. Saenko, Deep CORAL: correlation alignment for deep domain adaptation, in: G. Hua, H. Jégou (Eds.), Comput. Vis. – ECCV 2016 Work., Springer International Publishing, Cham, (2016) 443–450.
[25] G. Welch, G. Bishop, An introduction to the Kalman filter, In Pract. 7 (2006) 1–16.
[26] P. Kollmeyer, Panasonic 18650pf li-ion battery data, Mendeley Data. 1 (2018), https://doi.org/10.17632/wykht8y7tg.1.