Energy 201 (2020) 117664

Contents lists available at ScienceDirect

Energy

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

State-of-charge estimation of lithium-ion batteries using LSTM and UKF

Fangfang Yang [a], Shaohui Zhang [b], Weihua Li [c], Qiang Miao [b, *]

[a] School of Data Science, City University of Hong Kong, Hong Kong
[b] School of Mechanical Engineering, Dongguan University of Technology, Dongguan, 523808, China
[c] School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, 510641, China

ARTICLE INFO

Article history:
Received 8 August 2019
Received in revised form
27 March 2020
Accepted 18 April 2020
Available online 23 April 2020

Keywords:
Lithium-ion batteries
State-of-charge estimation
Ambient temperature
Recurrent neural network
Long short-term memory
Unscented kalman filter

ABSTRACT

For lithium iron phosphate battery, the ambient temperature and the flat open circuit voltage - state-of-charge (SOC) curve are two of the major issues that influence the accuracy of SOC estimation, which is critical for driving range estimation of electric vehicles and optimal charge control of batteries. To address these problems, this paper proposes a long short-term memory (LSTM) – recurrent neural network to model the sophisticated battery behaviors under varying temperatures and estimate battery SOC from voltage, current, and temperature variables. An unscented Kalman filter (UKF) is incorporated to filter out the noises and further reduce the estimation errors. The proposed method is evaluated using data collected from the dynamic stress test, federal urban driving schedule, and US06 test. Experimental results show that the proposed method can well learn the influence of ambient temperature and estimate battery SOC under varying temperatures from $0°C$ to $50°C$, with root mean square errors less than 1.1% and mean average errors less than 1%. Moreover, the proposed method also provides a satisfying SOC estimation under other temperatures which have no data trained before.

1. Introduction

Lithium-ion batteries have been recognized as the main energy storage device for electric vehicles (EVs) due to their extended cycle life and high energy/power density [1]. To ensure safe, reliable, and efficient operations of EVs, it is of utmost importance to build an advanced battery management system (BMS) to monitor the battery status accurately and timely. State-of-charge (SOC), defined as the ratio of remaining capacity over nominal capacity, indicates how much longer the battery can sustain before recharge is necessary [2]. Accurate knowledge of battery SOC is not only useful for EV range determination and trip planning, but also essential for the BMS to ensure battery working reliability and safety. Therefore, SOC is considered as one of the key characteristics in the BMS. However, it is not possible to measure battery SOC directly. In fact, it is still a challenging task to infer SOC from measurable signals such as terminal voltage, loading current and ambient temperature due to the highly sophisticated battery dynamics per se along with unpredictable operation conditions, such as regeneration, hysteresis, and cell aging [3].

Many approaches have been investigated for SOC estimation, including Coulomb counting approach, open-circuit voltage (OCV) approach, model-based filtering approach, and machine learning approach.

The Coulomb counting approach uses a current sensor to measure the discharge current and updates the battery SOC by subtracting the elapsed charge, which is the integral of discharge current over time [4]. While the Coulomb counting method is easy to implement, the estimation performance relies directly on the accuracy of initial battery SOC. Being essentially an open-loop estimator, it is not possible to correct the initialization error. Moreover, the estimation performance is also influenced by limited sensor precision and measurement noises. Practically, the Coulomb counting method can be used to measure the nominal capacity of batteries under lab condition.

The OCV approach approximates the SOC based on the unique OCV-SOC mapping [5]. This approach is not friendly to onboard applications, as accurate measurement of the OCV requires the battery in a steady state, which needs the battery be disconnected for sufficient time (often more than 2 h). Moreover, the OCV-SOC

* Corresponding author.
E-mail addresses: fangfyang2-c@my.cityu.edu.hk (F. Yang), zhangsh@dgut.edu.cn (S. Zhang), whlee@scut.edu.cn (W. Li), 2017831@dgut.edu.cn (Q. Miao).

https://doi.org/10.1016/j.energy.2020.117664
0360-5442/

relationship suffers from unit-to-unit variance and battery degradation. Finally, the estimation is very sensitive when the OCV-SOC curve is flat, in which case small errors of OCV induce great errors of SOC.

The model-based filtering approach considers SOC as a hidden state and constructs a state-space model that correlates SOC with measurable variables such as voltage and current. The SOC estimation can then be solved using extended Kalman filter (EKF), unscented Kalman filter (UKF), and particle filter (PF) [6]. The model-based filtering approach shows fair robustness against inaccurate SOC initialization and measurement noises. It turns out the performance of model-based filtering approach relies largely upon the accuracy of the underlying battery models. Many models have been proposed to improve the estimation performance, such as combined model, simple model, one-state hysteresis model, enhanced self-correcting model, temperature-based model, partial differential equation-based model, resistance-capacitance network based equivalent circuit model, etc. [7–10]. While these models yield satisfying estimations under strict conditions, such as specified battery type, constant ambient temperature, researchers are still in pursuit for a unified model that describes the sophisticated battery behaviors under real in-situ working condition.

The machine learning approach, such as fuzzy logic [11], support vector machine [12], Gaussian process regression (GPR) [13], and artificial neural network (NN) [14], sees the battery as a black box system, and models the nonlinear input-output relationship directly from large amounts of sample data. The machine learning approach requires no assumption on or explicit battery model and is extendible to other influencing factors. But its estimation accuracy counts on the quality and quantity of sample data and the training process is usually time-consuming.

Recently, the artificial NN-based methods are becoming increasingly popular among world-wide researchers. On one hand, the network training time is greatly shortened from days or even months to hours due to the ever-growing computing power from graphic processing units (GPUs) [15]. Meanwhile, huge amounts of training data can be gathered either by uploading field data from in-situ BMSs to remote data servers, or by conducting loading tests using various dynamic loading profiles under lab condition. What's more, the estimation can be done in few milliseconds with a pre-trained model, which is fast enough for most onboard EV applications. Chaoui et al. [16] proposed a recurrent NN (RNN) to estimate the SOC of lithium-ion batteries. It was shown that the RNN network shows strong robustness against dynamic loading, nonlinear dynamic nature, aging, hysteresis, and parametric uncertainties. By making use of historical information, the RNN presents better battery SOC estimation in comparison with conventional feedforward NNs.

Nonetheless, classic RNN is generally considered incapable of describing long-term dependences due to the gradient vanishing phenomenon, which invalidates the popular stochastic gradient-descent method used in training process [17]. To address this problem, an LSTM is proposed by Hochreiter and Schmidhuber [18]. In contrast to classic RNN, where gradients back-propagates exponentially, the LSTM allows the gradient to flow unchanged by employing a cell memory. By using an input gate, a forget gate, and an output gate, the LSTM unit gets to decide what to remember and what to forget and is therefore capable of addressing long-term dependencies. In practice, LSTMs have been successfully applied to battery SOC estimation such as Yang et al. [19], Song et al. [20], and Chemali et al. [21].

Besides the abovementioned issues, the complex battery systems suffer from the following two major problems on SOC estimation. One is the ambient temperature. Currently, most existing model-based filtering methods cannot be directly applied to copy with varying temperatures. Poor estimations have been reported for SOC at low temperatures [22]. Although some temperature-based models have been proposed to deal with the problems, such as the temperature-based internal resistance battery model proposed by Xing et al. [9] and the temperature-compensated model proposed by Liu et al. [23], they require to construct offline look-up tables under varying temperatures, which is operationally complex and suffer from unit-to-unit variance as well as battery aging. In comparison, the NN-based methods can learn to estimate the SOC under varying temperatures by simply adding the ambient temperature as an input factor, which is very easy to implement. The other major problem is the flat regions in the OCV-SOC curves, especially for the lithium iron phosphate (LFP) batteries [9]. This characteristic makes SOC estimations fluctuate, little voltage measurement error can lead to a large SOC estimation error. The flat OCV-SOC curve also adds difficulty on SOC estimation under varying temperatures.

To address these problems, a hybrid SOC estimator taking the advantages of the machine learning approach and the model-based filtering approach is proposed in this work to estimate the SOC of lithium-ion batteries. An LSTM-RNN network is employed to model the complex battery dynamics under varying temperatures and infer battery SOC from voltage, current, and temperature variables. A UKF is incorporated with the LSTM-RNN network to filter out the estimation noises from LSTM-RNN network. The performance of the proposed estimator is then evaluated with data collected from dynamic stress test, US06 test, and federal urban driving schedule under different temperatures. The major contributions of this study are threefold.

1) An LSTM-RNN network is trained to model the complex battery dynamics under varying ambient temperatures. A step-by-step searching method is presented to determine the optimal network hyper-parameters for SOC modeling. The network well learns the battery dynamics, presents good robustness against unknown initial states, and provides satisfying SOC estimations under varying temperatures.
2) An SOC estimator combining the LSTM-RNN network and the UKF filtering method is proposed. The UKF method is implemented to improve the SOC estimation of the LSTM-RNN network. The proposed estimator significantly improves the estimation accuracy during the flat OCV-SOC area and filters the estimation noises. The proposed method outperforms the other feed-forward machine learning methods in terms of both root mean square errors and mean absolute errors with acceptable computation time.
3) The proposed method is model-free and data-driven. There is no need to construct OCV-SOC lookup tables under varying temperatures and pick an appropriate equivalent circuit model. The proposed method should be able to be extended to the SOC estimation of other types of lithium-batteries.

The rest of the paper is organized as follows. An LSTM-RNN network is proposed for battery dynamic modeling in Section 2. The UKF-incorporated method to implement SOC estimation is demonstrated in Section 3. Section 4 illustrates the experiment design and data collection. Results and discussions are presented in Section 5. Finally, Section 6 concludes this paper.

2. Long short-term memory for battery dynamic modeling

In this section, LSTM, a variant of the RNN, is introduced to model the complex dynamics of lithium-ion batteries and infer battery SOC from measurable variables such as current, voltage, and temperature.

2.1. Recurrent neural network

Temporal relations are key for modeling sequential data, such as natural language processing, machine translation, and battery health estimation. Unlike traditional feed-forward NNs, which totally ignore the past inputs, the RNN [24] stores history information with recurrent units.

Fig. 1 demonstrates the structure of a recurrent unit *w*, where the hidden state h_k relates to not only the current input x_k but also the last hidden state h_{k-1}, which again relates to older inputs. The hidden state is updated recursively with

$$h_k = \begin{cases} 0, & k = 0 \\ \Phi([h_{k-1} x_k]), & \text{otherwise} \end{cases}, \tag{1}$$

where x_k is the input vector, and Φ is a nonlinear function, which often takes the following form:

$$h_k = \sigma_h(W_h x_k + U_h h_{k-1} + b_h), \tag{2}$$

where σ_h is a smooth, bounded activation function; W_h and U_h is are weight matrixes; b_h is the bias vector. The unit output y_k can be obtained as

$$y_k = \sigma_y(W_y x_k + U_y h_{k-1} + b_y). \tag{3}$$

As RNN naturally encodes history inputs, it excels at tasks with sequential input such as handwriting recognition and vocal translation. Nonetheless, classic RNN is incapable of handling long-term dependency, as the influence of past inputs fade out exponentially with time [17]. Besides, it also challenges the traditional gradient-based training methods as the gradients tend to vanish or rarely explode [17].

2.2. Long short-term memory network

To settle the gradient vanishing problem of the RNN, LSTM [18] designs a new activation function to weaken gradient vanishing rate and capture the long-term dependencies. The key idea of the LSTM is the cell memory, which streams down the entire chain subject to an affine transformation, if unpacked. Fig. 2 shows the structure of a typical LSTM unit, where c_k is the unit memory at time *k*. It relates to the previous unit memory by:

$$c_k = f_k \cdot c_{k-1} + \cdot i_k \cdot \tilde{c}_k, \tag{4}$$

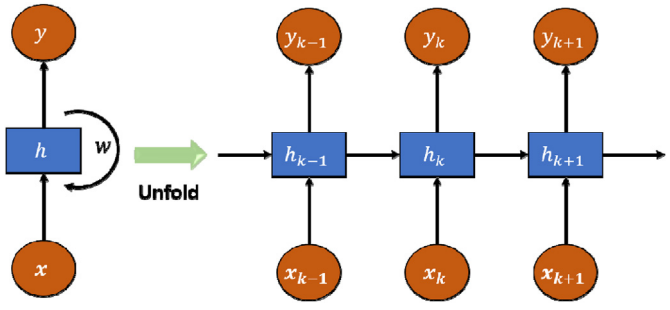

Fig. 1. Structure of the RNN unit.

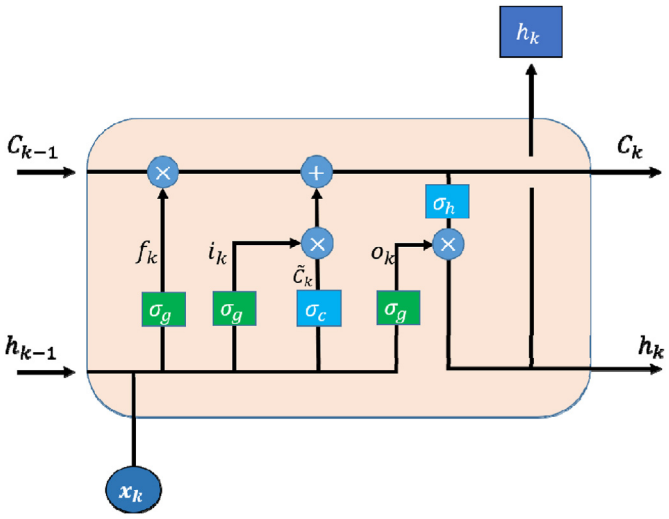

Fig. 2. Structure of the LSTM unit.

The LSTM unit utilizes three kinds of gate, i.e. the input gate *i*, the forget gate *f*, and the output gate *o*, to determine the proportion of the current cell memory to merge into the old cell memory, the forget rate of the cell memory upon current input, and the influence of the cell memory on the node output, respectively. The forward pass of an LSTM unit at time *k* is proceeded as follows:

$$\begin{aligned} f_k &= \sigma_g\left(W_f x_k + U_f h_{k-1} + b_f\right) \\ i_k &= \sigma_g(W_i x_k + U_i h_{k-1} + b_i), \\ o_k &= \sigma_g(W_o x_k + U_o h_{k-1} + b_o) \\ c_k &= f_k \circ c_{k-1} + i_k \circ \sigma_c(W_c x_k + U_c h_{k-1} + b_c) \\ h_k &= o_k \circ \sigma_h(c_k) \end{aligned} \tag{5}$$

where x_k and h_k are the unit input and corresponding output at time *k*, respectively; c_k denotes the hidden cell memory; *W*, *U*, and *b* are weight matrices and bias vectors to be learned during training; $\circ$ operator denotes the element-wise product between the vectors; i_k, f_k, and o_k represent the activation vectors of the input gate, the forget gate and the output gate, respectively, while σ_g, σ_c, σ_h are the corresponding activation functions; specifically, σ_g is a logistic sigmoid function while σ_c and σ_h are both hyperbolic tangent functions.

The sigmoid function is adopted as the gating function for the three gates because it outputs a value between 0 and 1, which shapes how the information gets through the gate. Take the forget gate for example, the resulting f_k is a vector with each element between 0 and 1. Intuitively, 0 means to be forgotten completely, while 1 means to be remembered exactly. Similarly, the input gate i_k determines the proportion of each element to be kept in $\tilde{c}_k$, which is then merged into the old cell memory. The same interpretation applies to the output gate analogously. In comparison, the hyperbolic tangent function is used to update the cell memory and the unit output, as the output of each neuron is constrained in the interval [−1,1], allowing for both increase and decrease of the cell memory and renormalization of the hidden state to [-1, 1]. Moreover, when the network is large, it is shown in Ref. [25] that the back-propagation learning with antisymmetric activation functions, such as the hyperbolic tangent function, can yield faster convergence than a similar process with non-symmetric activation functions, such as the sigmoid function.

Unlike RNN, where past inputs fade out exponentially, the LSTM controls the cell memory explicitly with forget gate and input gate. Old cell memory can be preserved after many temporal steps if the gating factor is close to 1. Meanwhile, errors can also flow backwards for many temporal steps, which greatly avoids gradient vanishing problem and enhances the network's capability of capturing long-term dependencies.

2.3. SOC modeling with LSTM-RNN network

As indicated by the Columbic counting method, there exists natural recursive relations between current SOC and past inputs. An LSTM-RNN network is therefore constructed to model the temporal dependencies, the architecture of which is shown in Fig. 3, where for the input layer, battery current, terminal voltage, and ambient temperature are used to form the input vector. To learn the temporal relations, one or more LSTM layers can be used, each unit of which is updated according to Equation (5). The output vector of the LSTM layers first goes through a fully connected layer and then an output layer for regression to produce the estimated SOC value. During the training process, the loss function evaluated at the end of each forward pass is mean squared error (MSE):

$$\mathrm{MSE}=\frac{1}{K}\sum_{k=1}^{K}(y_k-\hat{y}_k)^2, \tag{6}$$

where K is the length of the sequence; y_k is the true SOC value while $\hat{y}_k$ is the output of the proposed network at time k. A forward pass starts when the training data is fed into the network and ends when the SOC estimate is generated at each time step, and errors and overall loss are calculated simultaneously. In the backward pass, Adam optimizer [26] is adopted to minimize the total loss by updating the network weights and biases according to the gradient of the loss function. A training epoch usually consists of one or more batches. The training of each batch includes one forward pass and one backward pass, in which each sample in the batch has had an opportunity to update the internal network parameters. The forward and backward passes are continuously performed to update the network weights and biases until a convergence criterion is met. A dropout rate [27] of 20% is used in the hidden layers to prevent possible overtraining during the training phase. To prevent the training from diverging, before training, the training input [I, V, T] is normalized within the range of [-1 1] by using a min-max normalization [28]. Some of hyper-parameters in this work are initialized as: mini batch size: 60; initial learning rate: 0.01; gradient threshold: 1; decay rates: 0.9 and 0.999.

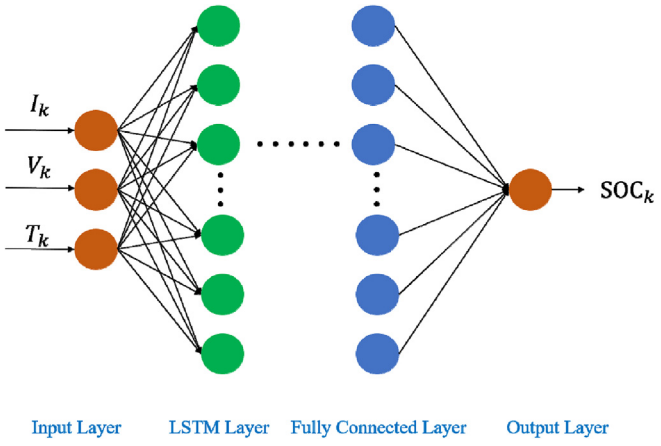

Fig. 3. Architecture of the LSTM-RNN network.

With a well-trained network, during the online testing process, only forward pass is required to produce the estimated SOC at each time step. The test input is normalized using the same parameters as the training input. Root mean squared error (RMSE) and mean absolute error (MAE) criteria are used to evaluate the estimation performance of the trained network:

$$\begin{aligned}\mathrm{RMSE}&=\sqrt{\frac{1}{K}\sum_{k=1}^{K}(y_k-\hat{y}_k)^2}.\\ \mathrm{MAE}&=\frac{1}{K}\sum_{k=1}^{K}|y_k-\hat{y}_k|\end{aligned} \tag{7}$$

The MAE measures how close estimates are to the corresponding outcomes. The RMSE, which characterizes the variation in errors, is more sensitive to large errors than the MAE.

3. Unscented kalman filter for battery SOC estimation

Variants of Kalman filter, as well-known techniques to strip noise out of a stream of data, are employed here to update a "best" estimate for the nonlinear battery systems. In this section, UKF is introduced to improve the estimation results from the LSTM-RNN. Compared with the EKF, the UKF shows better performance for nonlinear systems with same complexity class and is easier for implementation [29]. In addition, the EKF requires the derivatives of the state and measurement functions, which is not needed in the UKF approach. Thus, the LSTM-based battery model can be easily merged into the UKF to provide SOC estimations.

3.1. Unscented kalman filter

The key idea of the UKF [29] is to approximate the real system by unscented transformation [29,30], which is a numerical sampling technique deterministically finds a minimal set of sigma points to estimate the mean and variance of state variable under nonlinear transformation. The sigma points are then projected through the system equation to produce new sigma points. The newly estimated mean and covariance are formed based on their statistics. Consider propagating the system state $\mathbf{x}_k$with n- dimensional mean vector $\mathbf{x}_k^a$ and $n\times n$ covariance matrix $\mathbf{P}_k$ through a nonlinear function $\mathbf{z}_k=g(\mathbf{x}_k)$, the posterior statistics of z is determined by $2n+1$ sigma points x_k^i with corresponding weights W_i according to:

$$\begin{cases} x_{k-1}^0=\mathbf{x}_{k-1}^a \;\; i=0\\ x_{k-1}^i=\mathbf{x}_{k-1}^a+\sqrt{n+\lambda}\left[\sqrt{\mathbf{P}_{k-1}}\right]_i \;\; i=1,\ldots,n\\ x_{k-1}^{i+n}=\mathbf{x}_{k-1}^a-\sqrt{n+\lambda}\left[\sqrt{\mathbf{P}_{k-1}}\right]_{i-n} \;\; i=(n+1),\ldots,2n\\ W_0^m=\dfrac{\lambda}{\lambda+n} \;\; i=0\\ W_0^c=\dfrac{\lambda}{\lambda+n}+\left(1-\alpha^2+\beta\right)\\ W_i^m=W_i^c=\dfrac{1}{2(\lambda+n)} \;\; i=1,\ldots,2n \end{cases} \tag{8}$$

where $[\cdot]_i$ takes the ith column of the matrix; $\lambda=3\alpha^2-n$ controls

the spread of the sigma points around the mean; β relates to the distribution of $\mathbf{x}_{k-1}$. In this paper, α and β are empirically chosen as default values 1 and 0 [30], which means the spread of the sigma points x_{k-1}^{i} and x_{k-1}^{i+n} is $\sqrt{3}[\sqrt{\mathbf{P}_{k-1}}]_i$ away from the mean vector $\mathbf{x}_{k-1}^{a}$. The sigma points are projected through the nonlinear system process $z_k^{i,f} = g(x_k^{i,f}),\ i = 0,\ldots,2n$. The mean and covariance of the $\mathbf{z}_k^f$ are produced by:

$$\mathbf{z}_k^f = \sum_{i=0}^{2n} W_i^m z_k^{i,f}. \tag{9}$$

$$\mathrm{cov}\left(\mathbf{z}_k^f\right) = \sum_{i=0}^{2n} W_i^c \left(z_k^{i,f} - \mathbf{z}_k^f\right)\left(z_k^{i,f} - \mathbf{z}_k^f\right)^T. \tag{10}$$

The implementation details of UKF [29,31] for state estimation are summarized in Table 1. In general, the UKF has a prediction-update structure. For each time step k, the state mean and covariance are first predicted based on the state at step k-1; and then in the update step, the predicted mean and covariance are updated based on new measurements obtained at current time step.

3.2. SOC estimation based on the proposed method

In the SOC estimation, the state vector is $\mathbf{x} = [\mathrm{SOC}]$, it follows Coulomb counting to derive the state function. The measurement vector is the SOC outputs of the LSTM-RNN network. The LSTM-RNN network is chosen as the measurement function. The UKF is performed to filter out the noises in the LSTM-RNN observations and update the SOC estimation. The state-space model is as follows:

Table 1
The UKF procedure for state estimation [29,31].

- Initialization:

≻Initial states, $\boldsymbol{x}_0^a = \mathrm{E}[\boldsymbol{x}_0]$ with covariance, $\boldsymbol{P}_0 = E[(\boldsymbol{x}_0 - \boldsymbol{x}_0^a)(\boldsymbol{x}_0 - \boldsymbol{x}_0^a)^{\mathrm{T}}]$

- For $k = 1, 2, \cdots$

≻Derive sigma points:

$\boldsymbol{x}_{k-1}^f = [\boldsymbol{x}_{k-1}^a\ \boldsymbol{x}_{k-1}^a + \sqrt{n+\lambda}[\sqrt{\boldsymbol{P}_{k-1}}]_i\ \boldsymbol{x}_{k-1}^a - \sqrt{n+\lambda}[\sqrt{\boldsymbol{P}_{k-1}}]_i]$

≻State prediction

- Project the sigma points through the state function:

$\boldsymbol{x}_k^{i,f} = \mathbf{f}(\boldsymbol{x}_{k-1}^i)$

- Compute the mean and covariance of the forecast state:

$\boldsymbol{x}_k^f = \sum_{i=0}^{2n} W_i^m \boldsymbol{x}_k^{i,f}$

$\boldsymbol{P}_k^f = \sum_{i=0}^{2n} W_i^c (\boldsymbol{x}_k^{i,f} - \boldsymbol{x}_k^f)(\boldsymbol{x}_k^{i,f} - \boldsymbol{x}_k^f)^{\mathrm{T}}$

≻Measurement update

- Project the sigma points through the measurement function:

$\boldsymbol{z}_k^{i,f} = \mathbf{h}(\boldsymbol{x}_k^{i,f})$

- Compute the mean and covariance of the measurement:

$\boldsymbol{z}_k^f = \sum_{i=0}^{2n} W_i^m \boldsymbol{z}_k^{i,f}$

$\mathrm{cov}(\boldsymbol{z}_k^f) = \sum_{i=0}^{2n} W_i^c (\boldsymbol{z}_k^{i,f} - \boldsymbol{z}_k^f)(\boldsymbol{z}_k^{i,f} - \boldsymbol{z}_k^f)^{\mathrm{T}}$

- Compute the cross-covariance between the state and the measurement:

$\mathrm{cov}(\boldsymbol{x}_k^f, \boldsymbol{z}_k^f) = \sum_{i=0}^{2n} W_i (\boldsymbol{x}_k^{i,f} - \boldsymbol{x}_k^f)(\boldsymbol{z}_k^{i,f} - \boldsymbol{z}_k^f)^{\mathrm{T}}$

- Calculate the Kalman gain K and update the state estimation and covariance:

$\boldsymbol{K}_k = \mathrm{cov}(\boldsymbol{x}_k^f, \boldsymbol{z}_k^f)\mathrm{cov}(\boldsymbol{z}_k^f)^{-1}$

$\boldsymbol{x}_k^a = \boldsymbol{x}_k^f + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{z}_k^f)$

$\boldsymbol{P}_k = \boldsymbol{P}_k^f - \boldsymbol{K}_k \mathrm{cov}(\boldsymbol{z}_k^f)\boldsymbol{K}_k^{\mathrm{T}}$

State function:

$$\mathrm{SOC}_k = \mathbf{f}(\mathrm{SOC}_{k-1}) = \mathrm{SOC}_{k-1} - \frac{I_{k-1} \times \Delta t}{\mathrm{C}_n} + \omega, \tag{11}$$

Measurement function:

$$\mathrm{LSTM}_k = \mathbf{h}(\mathrm{SOC}_k) = \mathrm{SOC}_k + \upsilon, \tag{12}$$

where I_k is the current at time k, C_n is the nominal capacity, Δt is the sampling interval, and $\omega \sim N(0, \mathrm{Q})$ and $\upsilon \sim N(0, \mathrm{R})$ are the Gaussian state noise and Gaussian measurement noise, respectively. Here Q and Rare set to 10^{-5} and 10^{-2}, respectively, by considering the uncertainties and scales of the state and the terminal voltages.

Fig. 4 illustrates the proposed battery SOC estimation framework, which consists of an offline training stage and an online testing stage. In the offline stage, the LSTM-RNN network is trained with collected data, which can be either field data or experimental data. In the online stage, a UKF is appended directly after the output of the well-trained network to filter the unwanted noises and improve the estimation results.

4. Experiments design and data collection

In the experiments, cylindrical 18650 lithium iron phosphate batteries from A123 Systems, LLC, were used, the key parameters of which are tabulated in Table 2. To better simulate real-life EV battery loading behaviors, three dynamic loading profiles, including dynamic stress test (DST), US06 test, and federal urban driving schedule (FUDS) were employed. All these profiles were designed by the US Advanced Battery Consortium to simulate different discharge patterns of an EV battery. Specifically, the FUDS profile focuses on EV battery usage under city driving conditions, while US06 profile corresponds to highway driving conditions. The DST profile has the same average characteristics as FUDS with only seven discrete power levels. Details of the DST, US06 and FUDS profiles are drawn in Fig. 5(a). In this paper, the data collected from DST and US06 tests are used to train the network while those from FUDS test serve as the testing dataset.

The overall test flow is shown in Fig. 6. To take ambient temperature into consideration, the test process was conducted under 0°C, 10°C, 20°C, 30°C, 40°C, 50°C, and room temperature (RT, around 27°C) using a temperature chamber from Votsch. We chose the temperature range with refer to Ref. [9], in the hope of covering a broad working range [32] a battery may undergo. To fully reinstate the battery, there was 3-h gap before starting the discharge test. During the whole tests, battery current, terminal voltage, ambient temperature were recorded every 1 s. Fig. 5(b) shows the measured voltages at RT for the DST, US06, and FUDS tests, respectively.

For each discharge test, the battery was first charged using constant-current (1C) and constant-voltage (3.6 V, as in Table 2) charging profile. The charge process was terminated when the current dropped below the end-of-charge current (0.01C). In the discharge process, which tool off right after the end of charge process, one of the abovementioned discharge profiles was applied recurrently to simulate different battery loading behaviors until fully discharge. The battery was considered fully discharged when the terminal voltage dropped below the discharge cut-off voltage (2 V).

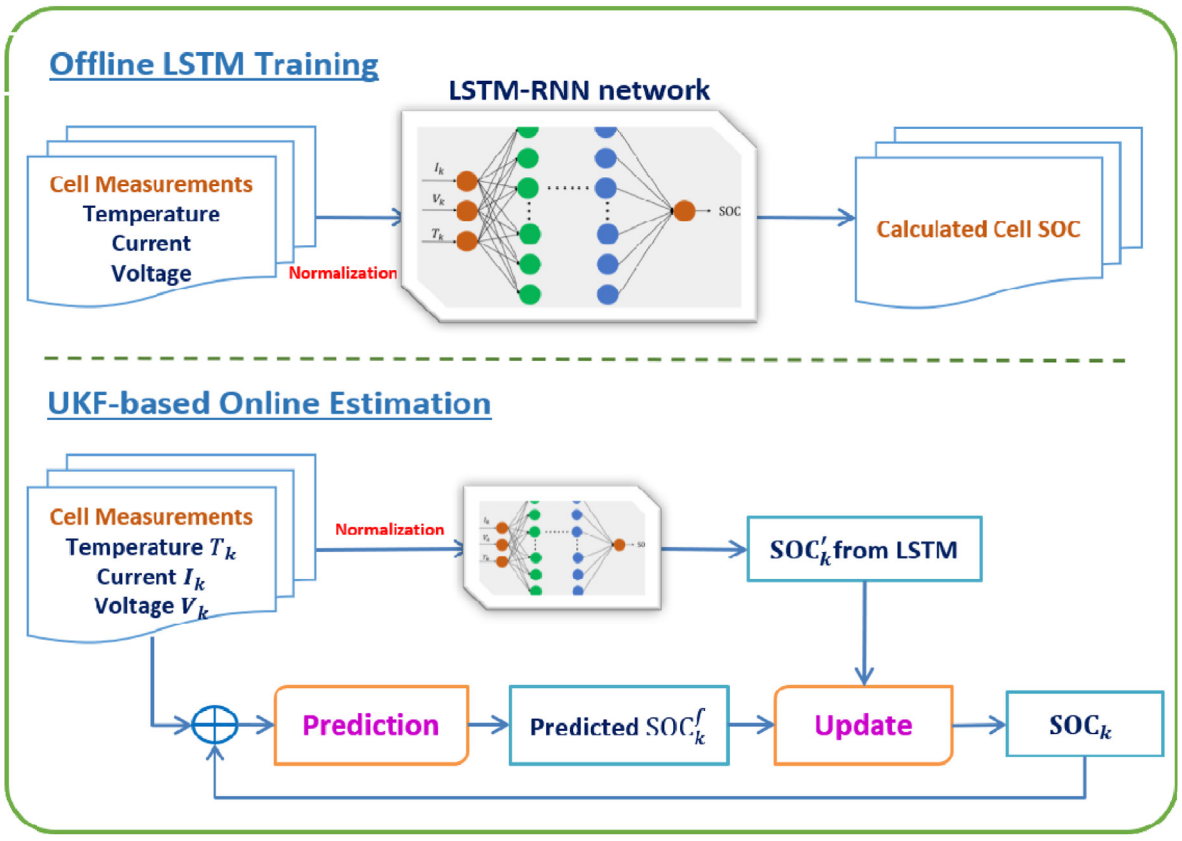

Fig. 4. Framework of the SOC estimation.

Table 2
Specifications of the subject batteries.

Type	A123 18650
Cathode	LFP
Anode	Graphite
Nominal capacity	1.1Ah
Charge voltage	3.6 V
Discharge cut-off voltage	2 V
End-of-charge current	0.01C

5. Estimation results

In this section, data collected in Section 4 are used to demonstrate the effectiveness of the proposed approach. The determination of the LSTM-RNN network is presented in Section 5.1. The estimation results under varying temperatures are discussed in Section 5.2.

5.1. Determination of the network parameters

In this section, an LSTM-RNN network is trained for battery SOC estimation with data collected at room temperature from the DST test (7200 samples) and the US06 test (7100 samples). The network takes battery current, terminal voltage, and ambient temperature as network input and outputs the SOC at the same cycle. Data collected from the FUDS test (7400 samples) are used to evaluate the performance of the network. Key hyper-parameters including the epoch number, the hidden neuron, and the hidden layer are determined step by step as following. The offline training was done using our lab server with two GeForce GTX 1080Ti GPUs.

First, a 300-neuron structure in the hidden LSTM layer is used. Fig. 7 shows the training and testing performance under different epoch numbers. In general, training and testing errors reduce with more epoch number available, i.e., longer the training time. Trading-off between the testing accuracy and training cost, 8000 is a reasonable epoch number, which takes about 1.7 h under our hardware environment. During the testing, average computation time of each network forward pass is 0.064 ms on our lab computer (CPU: Intel i7-7500U @2.70 GHz, Memory: 8 GB, OS: Win10 x64,), which meets the requirement of real-time onboard applications.

Next, varying LSTM-neurons are tested to determine an appropriate neuron number. Fig. 8 shows the training and testing RMSEs with LSTM-neuron number varying among [50 100 200 300 400]. While using more neurons reduces the training RMSE, too many of which tend to over-fit, which features a small training error but a large testing error. As a trade-off, 300 is considered as the optimal LSTM-neuron number.

Last, the effect of additional LSTM layers is studied. A set of networks with different LSTM layer configurations are constructed for performance comparison. Network details and their estimation performance are listed in Table 3. The optimal epoch number is determined using the same method as described above. The results show that single LSTM layer (network 1) yields comparable results with double LSTM layers, such as networks 2 and 3, and has a better performance than triple LSTM layers.

It is important that the proposed network can still work well in case the initial SOC is unknown, which is very common-seen in practice. For this purpose, discharge data at varying initial SOCs are generated by truncating the full-discharge data at the closest point. In this work, initial SOCs at 80%, 60% and 40% are generated. Fig. 9

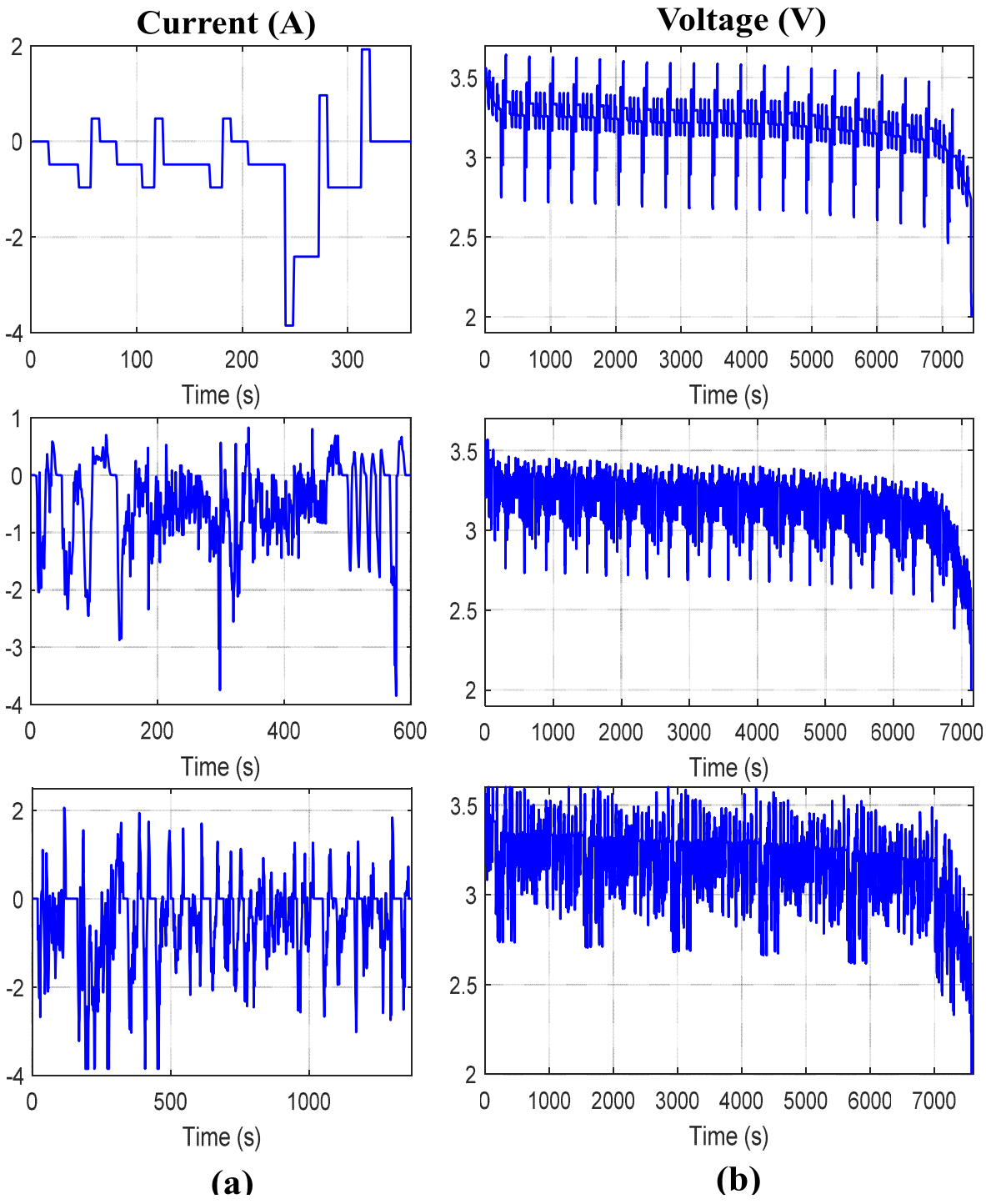

Fig. 5. Current profile (a) and measured voltage (b) of DST (top row), US06 (middle row) and FUDS (bottom row) in one discharge cycle.

plots the estimation results of network 1, 2, and 3, where the estimation starts from four initial states (100%, 80%, 60%, and 40%). All the networks can converge to true SOC even when initial SOC are unknown. Compared with cases from other starting SOCs, estimation from 100% SOC has no starting deviation issue, as all input sequences start from 100% SOC and end at 0% SOC in the training process. When SOC starts from 80%, 60%, and 40%, there are some estimation deviations at starting points. As the LSTM relates current estimate with past inputs, the estimate converges to true values gradually. As in Fig. 9(b), network 1 and 2 show better performance than network 3. In all cases, slightly larger errors are observed during the 80%–40% SOC regions, which arises from the existence of the flat region in the OCV-SOC curve of the LFP battery. See Section 5.2 for details. During this region, network 1 presents more accurate estimation results than network 2. Thus, single LSTM layer with 300 neurons is deployed in the following experiments. One trade-off to decrease the starting variations is to include truncated training input sequences at varying starting SOCs for network training, which would cost more training time.

In this section, a simple yet effective procedure is presented to

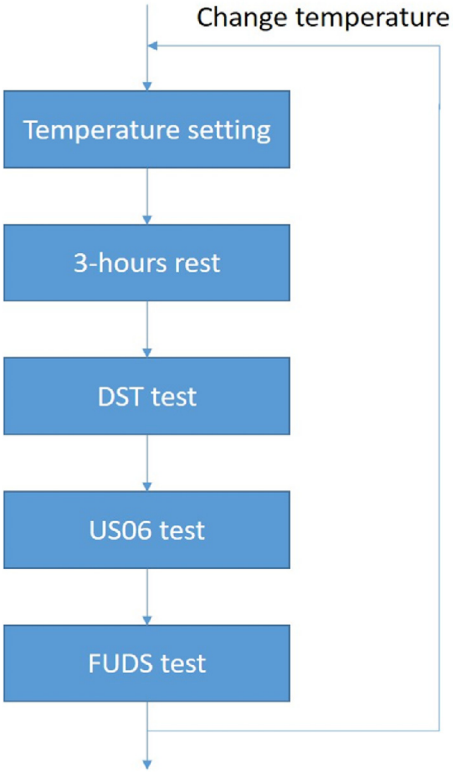

Fig. 6. Test procedure for the battery samples.

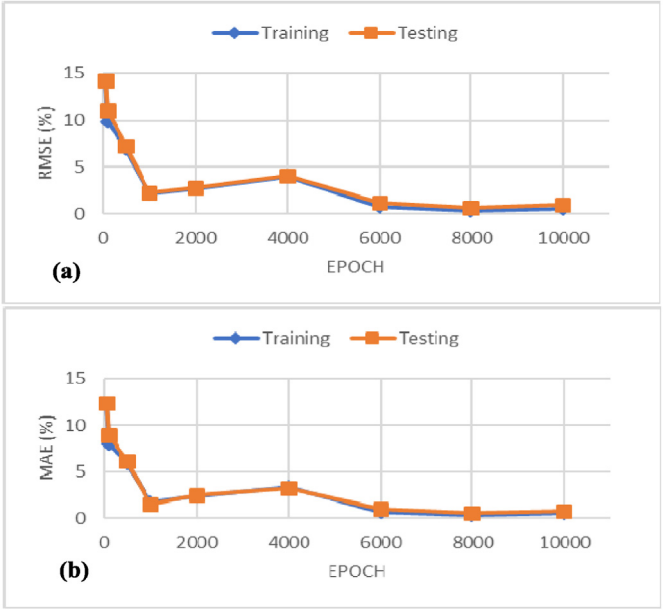

Fig. 7. Training and testing RMSEs (a) and MAEs (b) vs. epoch number.

determine the appropriate network parameters. For applications with different battery materials or a larger data size, these procedures can be carried out in a similar way to determine the corresponding network parameters.

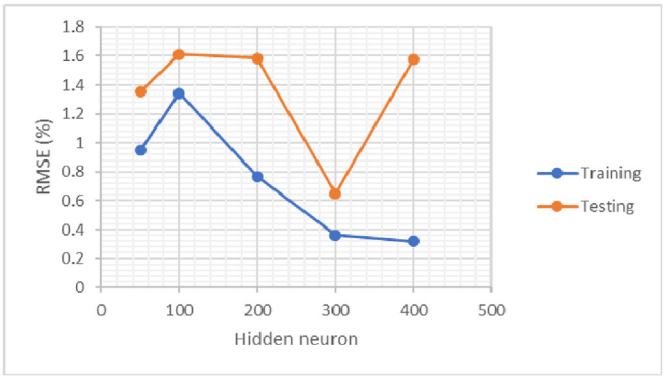

Fig. 8. Training and testing RMSEs vs. number of hidden neurons.

5.2. SOC estimation under varying temperatures

To capture the effects of ambient temperature, the proposed network is trained to estimate the battery SOC under varying temperatures in this section. A UKF filtering technique is used to further enhance the performance of the LSTM-RNN network.

In the offline stage, the proposed network is trained with data collected from DST and US06 tests at 0°C, 10°C, 20°C, 30°C, 40°C, and 50°C. Data at RT is left out in order to test the extrapolation ability of the proposed network. The whole training process took 2.6 h with 10000 training epochs. Performance of the network is evaluated with data collect from FUDS test under 0°C, 10°C, 20°C, 30°C, 40°C, 50°C, and RT, respectively.

Fig. 10 shows the SOC estimation at 10°C, RT, and40°C, respectively. Although the network is not trained with data at RT, the proposed LSTM-RNN network yields satisfying SOC estimation, as in Fig. 10(b). The estimated SOC (blue dot dash line) approaches the true SOC, with a RMSE of 1.93% and a MAE of 1.58%, which is quite satisfactory. Nonetheless, a relatively large estimation error is observed during 80%–40% SOC. The same phenomenon is observed in Fig. 10(c), where the SOC is estimated under 40°C. As a comparison, the network presents a better estimation at low temperatures (0°C, 10°C), especially during the 80%–40% SOC, as in Fig. 10(a), the corresponding RMSE and MAE are 1.06% and 0.88%, respectively. This can be explained by the existence of flat region in the OCV-SOC curve for the LFP batteries. Fig. 11 shows the typical OCV-SOC curves for the LFP batteries under varying ambient temperatures. For a fixed temperature, it is clearly that the middle region of the curve is relatively flat, which means that a small error in voltage would result in a large SOC deviation. Moreover, the figure also indicates that the higher the temperature, the flatter the curve in the middle region. In other words, for the same error in the voltage measurement, a larger deviation of the estimated SOC is induced at a high temperature.

To account for this phenomenon, a UKF is appended to further filter the network output. The estimation results using the LSTM-UKF approach are presented in red dash lines. For SOC estimation under RT and high temperature, as shown in Fig. 10(b–c), the proposed method significantly improves the estimation during the 80%–40% SOC. For SOC estimation under RT, the corresponding RMSE and MAE are 1.06% and 0.93%, respectively. Compared with the estimation by the LSTM-RNN network, there is a reduction of 45% and 41%. For SOC estimation under low temperature in Fig. 10(a), the RMSE is reduced from 1.43% to 0.73% while the MAE is reduced from 1.31% to 0.63%.

Table 3
Comparison of stacked LSTM-RNN networks. '25', '50', '100', and '300' refer to the number of hidden neurons in the LSTM layers.

Network	Model	Training				Testing	
		RMSE (%)	MAE (%)	epoch	Time (h)	RMSE (%)	MAE (%)
1	LSTM 300	0.36	0.31	8000	1.7	0.65	0.52
2	LSTM 50-50	0.49	0.42	10000	1.8	0.89	0.64
3	LSTM 100-50	0.50	0.43	10000	1.8	0.53	0.42
4	LSTM 100-100	0.78	0.59	10000	1.8	0.88	0.72
5	LSTM 25-25-25	0.70	0.63	10000	1.9	1.10	0.93
6	LSTM 50-50-50	0.48	0.40	10000	1.9	1.07	0.84

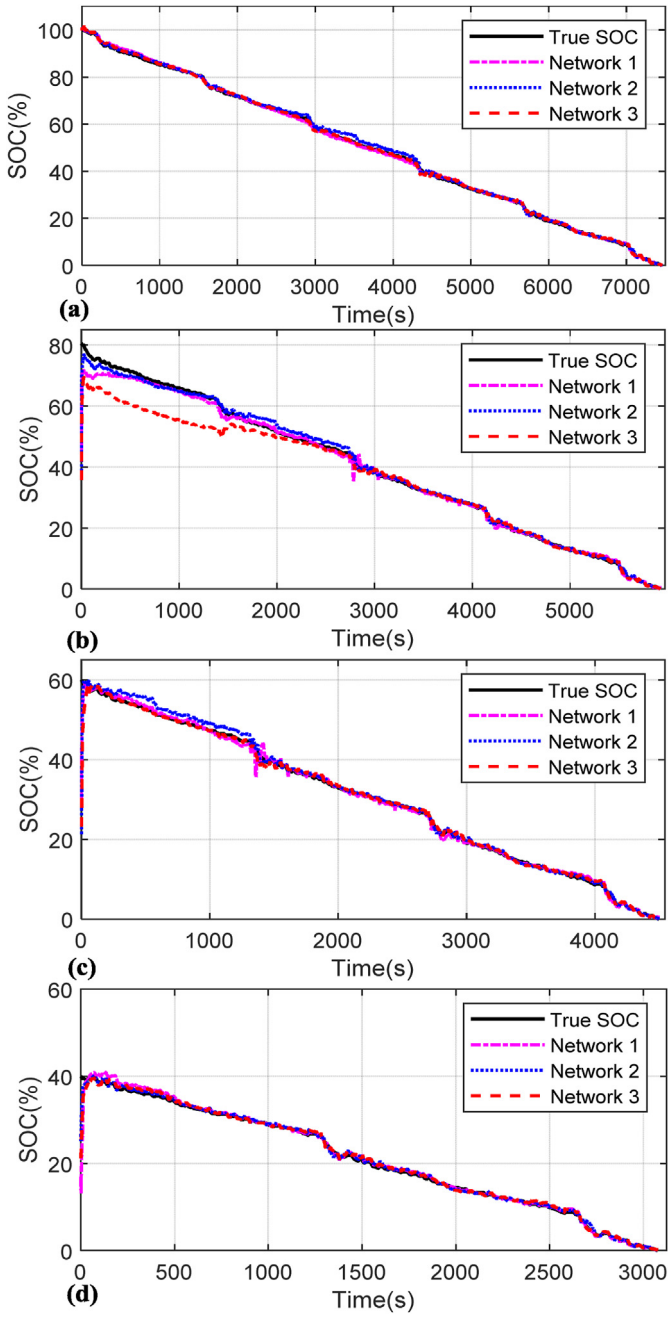

Fig. 9. SOC estimation with different LSTM-RNN networks from: (a) 100% SOC; (b) 80% SOC; (c) 60% SOC; (d) 40% SOC.

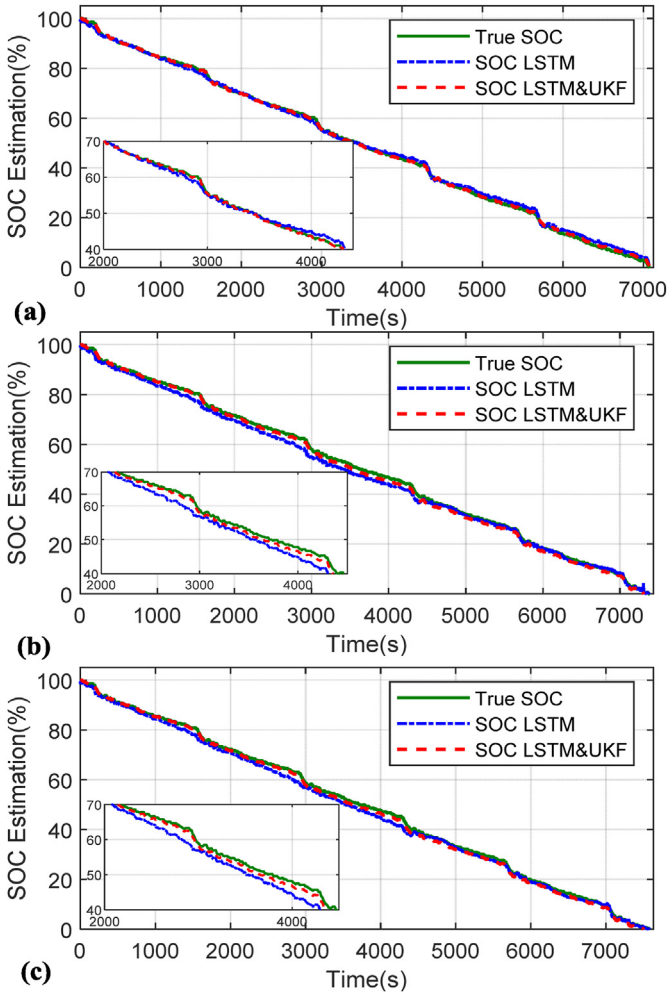

Fig. 10. SOC estimation at different temperatures: (a)10°C; (b) RT; (c)40°C.

Table 4 tabulates the comparison of the LSTM-RNN network with and without a UKF filter in all cases. Overall, the LSTM-RNN network yields quite satisfying SOC estimation, with RMSEs and MAEs less than 2.1% and 1.6%, respectively. Better results are obtained by using the UKF filtering, where both RMSEs and MAEs are reduced to within 1.1% and 1%, respectively. Hence, we conclude that the LSTM-RNN network can capture the influence of ambient temperature and can produce satisfying results at untrained temperatures. Moreover, the performance of SOC estimation can be

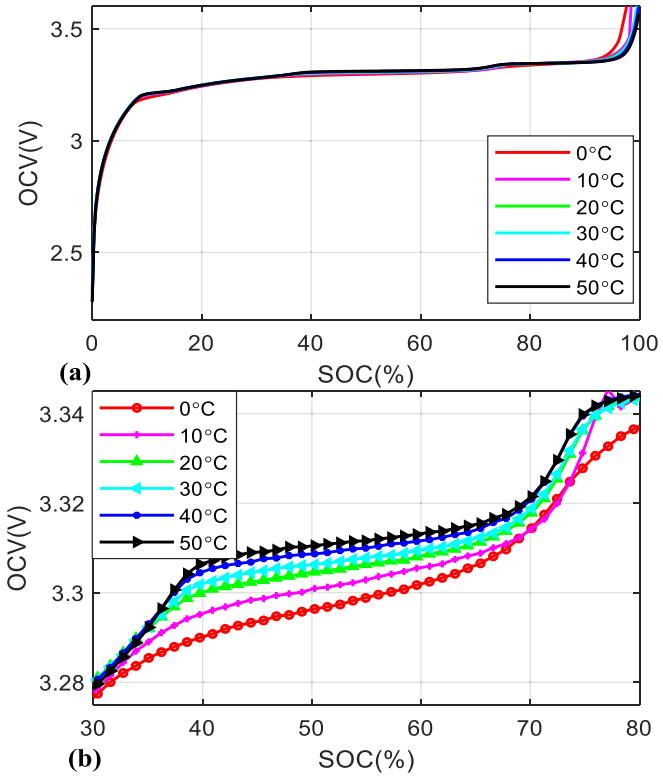

Fig. 11. OCV-SOC curves at different temperatures: (a) 0%–100% SOC; (b) 30%–80% SOC.

Table 4
RMSEs and MAEs of SOC estimation under varying temperatures.

Temperature (°C)	RMSE (%)		MAE (%)	
	LSTM	LSTM&UKF	LSTM	LSTM&UKF
0	1.46	0.73	1.31	0.63
10	1.06	0.29	0.88	0.21
20	2.04	1.11	1.60	0.97
30	1.68	0.93	1.40	0.82
40	1.68	0.92	1.38	0.81
50	1.86	1.03	1.51	0.89
RT	1.93	1.06	1.58	0.93

further improved with a UKF filter.

In reality, the operating temperature of battery could range from −30°C to 60°C [33], which is not fully covered in our experiments. It has been reported that the battery dynamics at low temperatures are much more complicated [22]. As reflected in Refs. [34], the voltage-capacity curves below 0°C are more complex than those of other higher temperatures. Subsequently, SOC estimation at lower temperatures is much more difficult. Considering the possible bad performance under low temperatures, to improve the estimation, one practical solution is to train networks particularly for battery working at low temperatures. Under the premise of sufficient data, the proposed framework should yield satisfying estimation results as well.

5.3. Performance comparison with other methods

In this section, the estimation performance of proposed method is compared with other popular machine learning methods including SVM, GPR, and NN in terms of estimation error and computation time. Compared with filtering methods, these methods can easily handle estimation at varying temperatures, with no need to construct OCV-SOC-temperature table or train model separately at different temperatures. To optimize the estimation performance, the SVM is tested using linear, Gaussian, and polynomial kernel of different orders with data standardization, while the GPR is tested using rational quadratic, squared exponential, matern 5/2, and exponential kernel, with data standardization. As a result, Gaussian kernel (scale: 0.43) and exponential kernel function are chosen for SVM method and GPR method, respectively. For NN, we employ the same structure as the LSTM. The input and output are the same as before. Finally, 5-fold cross-validation is used to optimize the model parameters for estimation performance.

Fig. 12 shows the estimation results on FUDS data at room temperature (with DST and US06 data at 0°C, 10°C, 20°C, 30°C, 40°C, and 50°C being trained). Table 5 tabulates the corresponding RMSE, MAE values, and computation time for the whole estimation process. Compared with NN and SVM, the GPR shows slightly better performance in terms of RMSE (7.3%) and MAE (5%) at the cost of more computation time. The three methods are robust against inaccurate SOC initials due to their feed-forward structures and that the output solely relies on the current status. For these methods, the estimated SOCs fluctuate seriously along true values, with RMSEs larger than 7% and MAEs larger than 5%. This may because of both the flat OCV-SOC nature (Fig. 11) and the dynamic current change in FUDS data. In FUDS profile, the discharge current changes drastically every second (Fig. 5). In contrast, the proposed method provides much better estimation performance with RMSE of 0.93% and MAE of 0.82%, when SOC starts from 100%. When SOC estimated from 80%, the proposed method converges fast to the true value within 100 s. The computation time of proposed method is slightly larger than those of NN and SVM and smaller than that of GPR.

Similar estimation performance is observed at other temperatures, the results are thus omitted here. Overall, the proposed method outperforms the other feed-forward machine learning methods in terms of both RMSEs and MAEs with acceptable computation time.

6. Conclusion

In this paper, a hybrid SOC estimator combining long short-term memory – recurrent neural network and unscented Kalman filter was proposed to estimate the SOC of lithium iron phosphate batteries under different temperatures. The network takes current, voltage and temperature as input, and outputs the SOC estimation. A step-by-step searching strategy was presented to find the optimal network structure for SOC estimation. Experimental results showed that the network well learned the battery dynamics and provided a satisfying SOC estimation, even under untrained temperature conditions. Besides, the network showed a good

Table 5
Comparison results of SOC estimation under room temperature.

Methods	RMSE (%)	MAE (%)	Computation time (s)
Proposed method	0.93	0.82	1.20
NN	8.13	5.72	0.039
SVM	8.80	5.85	0.57
GPR	7.30	4.99	2.32

robustness against unknown initial states. A UKF was incorporated to filter out the noises in the network output and improve the SOC estimation accuracy during the flat OCV-SOC regions. A set of dynamic loading profiles were used to generate the training and testing datasets, including the dynamic stress test, US06 test, and federal urban driving schedule. Experimental results showed the proposed estimator provided good SOC estimations under varying temperatures, with RMSEs within 1.1% and MAEs within 1%. Besides, the proposed method outperformed the other feed-forward machine learning methods in terms of both RMSEs and MAEs with acceptable computation time.

The proposed method is model-free and data-driven. There is no need to construct OCV-SOC lookup tables under varying temperatures and pick an appropriate equivalent circuit model. Subsequently, the proposed method should be able to be extended to the SOC estimation of other types of lithium-batteries, such as the lithium nickel manganese cobalt oxide battery.

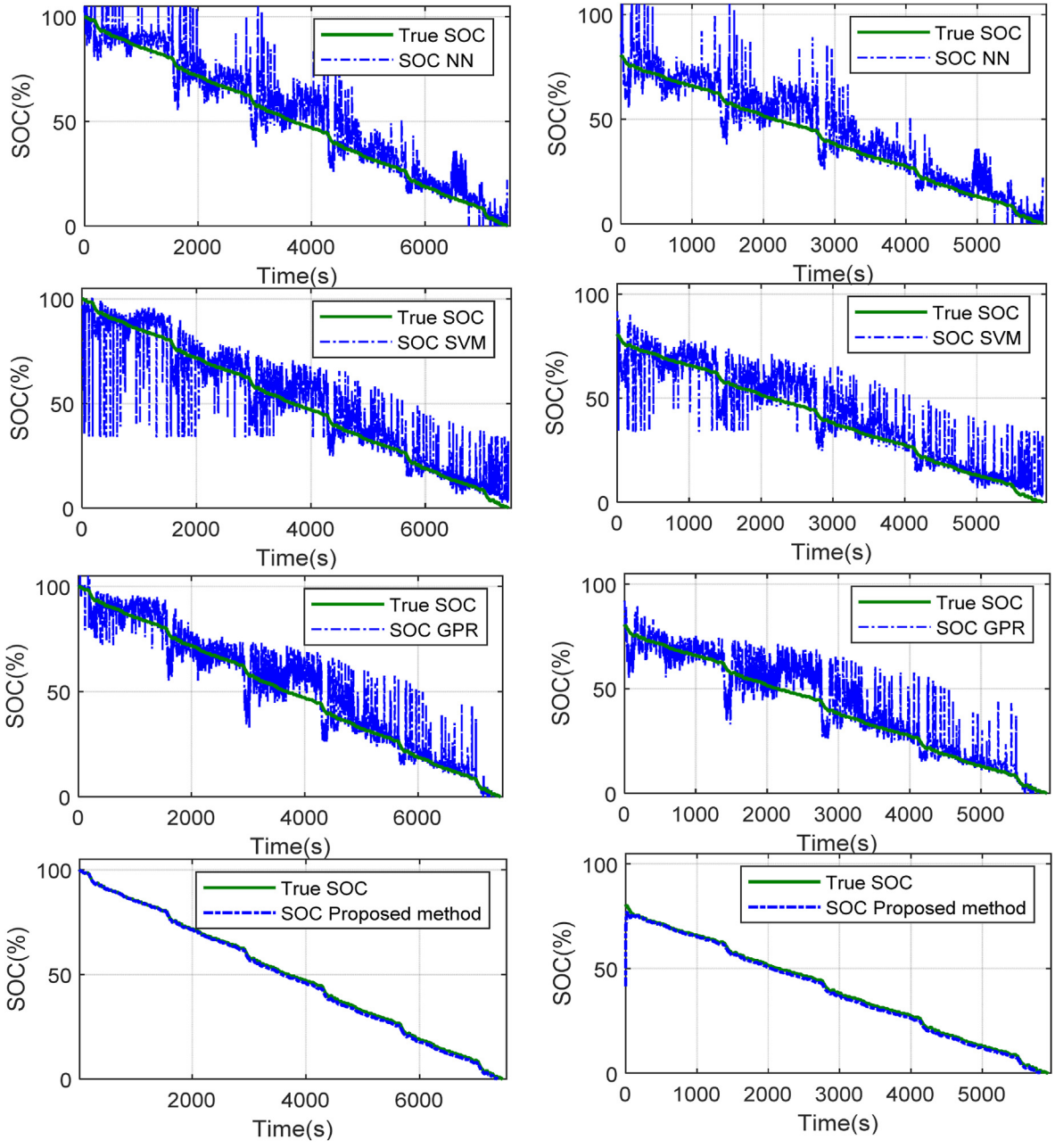

Fig. 12. SOC estimation when SOC starts from 100% (left column) and 80% (right column) at room temperatures. (a–b): NN; (c–d): SVM; (e–f): GPR; (g–h): Proposed method.

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.

CRediT authorship contribution statement

Fangfang Yang: Conceptualization, Data curation, Methodology, Software, Validation, Writing - original draft. **Shaohui Zhang:** Investigation, Software, Validation. **Weihua Li:** Supervision, Writing - review & editing. **Qiang Miao:** Project administration, Supervision, Writing - review & editing.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (51875208, 51475170) and the DGUT Grant (GC300501-27).

References

[1] Yang F, Wang D, Zhao Y, Tsui K-L, Bae SJ. A study of the relationship between coulombic efficiency and capacity degradation of commercial lithium-ion batteries. Energy 2018;145:486–95.
[2] Hannan MA, Lipu MH, Hussain A, Mohamed A. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: challenges and recommendations. Renew Sustain Energy Rev 2017;78:834–54.
[3] Lu L, Han X, Li J, Hua J, Ouyang M. A review on the key issues for lithium-ion battery management in electric vehicles. J Power Sources 2013;226:272–88.
[4] Ng KS, Moo C-S, Chen Y-P, Hsieh Y-C. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. Appl Energy 2009;86(9):1506–11.
[5] 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(1):310–8.
[6] 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;164:387–99.
[7] Plett GL. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification. J Power Sources 2004;134(2):262–76.
[8] Hu X, Li S, Peng H, Sun F. Robustness analysis of State-of-Charge estimation methods for two types of Li-ion batteries. J Power Sources 2012;217:209–19. 0.
[9] Xing Y, He W, Pecht M, Tsui KL. State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures. Appl Energy 2014;113:106–15. 0.
[10] Zou C, Manzie C, Nešić D. A framework for simplification of PDE-based lithium-ion battery models. IEEE Trans Contr Syst Technol 2016;24(5):1594–609.
[11] Salkind AJ, Fennie C, Singh P, Atwater T, Reisner DE. Determination of state-of-charge and state-of-health of batteries by fuzzy logic methodology. J Power Sources 1999;80(1–2):293–300.
[12] Anton JA, Nieto PG, Viejo CB, Vilan JV. Support vector machines used to estimate the battery state of charge. IEEE Trans Power Electron 2013;28(12):5919–26.
[13] Sahinoglu GO, Pajovic M, Sahinoglu Z, Wang Y, Orlik PV, Wada T. Battery state-of-charge estimation based on regular/recurrent Gaussian process regression. IEEE Trans Ind Electron 2018;65(5):4311–21.
[14] Kang L, Zhao X, Ma J. A new neural network model for the state-of-charge estimation in the battery degradation process. Appl Energy 2014;121:20–7. 0.
[15] Yang F, Li W, Li C, Miao Q. State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network. Energy; 2019.
[16] Chaoui H, Ibe-Ekeocha CC. State of charge and state of health estimation for lithium batteries using recurrent neural networks. IEEE Trans Veh Technol 2017;66(10):8773–83.
[17] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Network 1994;5(2):157–66.
[18] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[19] Yang F, Song X, Xu F, Tsui K-L. State-of-Charge estimation of lithium-ion batteries via long short-term memory network. IEEE Access 2019;7:53792–9.
[20] Song X, Yang F, Wang D, Tsui K-L. Combined CNN-LSTM network for state-of-charge estimation of lithium-ion batteries. IEEE Access 2019;7:88894–902.
[21] Chemali E, Kollmeyer PJ, 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(8):6730–9.
[22] Waldmann T, Wilka M, Kasper M, Fleischhammer M, Wohlfahrt-Mehrens M. Temperature dependent ageing mechanisms in Lithium-ion batteries–A Post-Mortem study. J Power Sources 2014;262:129–35.
[23] Liu X, Chen Z, Zhang C, Wu J. A novel temperature-compensated model for power Li-ion batteries with dual-particle-filter state of charge estimation. Appl Energy 2014;123:263–72.
[24] Liu P, Qiu X, Huang X. Recurrent neural network for text classification with multi-task learning. 2016. arXiv preprint arXiv:1605.05101.
[25] Kalman BL, Kwasny SC. Why tanh: choosing a sigmoidal function. In: [Proceedings 1992] IJCNN international joint conference on neural networks. IEEE; 1992. p. 578–81.
[26] Kingma DP, Ba J, Adam. A method for stochastic optimization. 2014. arXiv preprint arXiv:1412.6980.
[27] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–58.
[28] Jain YK, Bhandare SK. Min max normalization based data perturbation method for privacy protection. International Journal of Computer & Communication Technology 2011;2(8):45–50.
[29] Julier SJ, Uhlmann JK. Unscented filtering and nonlinear estimation. Proc IEEE 2004;92(3):401–22.
[30] S. Särkkä, Bayesian filtering and smoothing, Cambridge University Press2013.
[31] Julier SJ, Uhlmann JK. New extension of the Kalman filter to nonlinear systems, Signal processing, sensor fusion, and target recognition VI. International Society for Optics and Photonics; 1997. p. 182–93.
[32] Hunt G. USABC electric vehicle battery test procedures manual. Washington, DC, USA: United States Department of Energy; 1996.
[33] A123 APR18650m1A cell, https://www.batteryspace.com/prod-specs/6612.pdf.
[34] Feng F, Lu R, Zhu C. A combined state of charge estimation method for lithium-ion batteries used in a wide ambient temperature range. Energies 2014;7(5):3004–32.