Proceedings of the 39th Chinese Control Conference
July 27-29, 2020, Shenyang, China

Battery State of Charge Estimation Using Long Short-Term Memory Network and Extended Kalman Filter

Zichuan Ni, Ying Yang*, and Xianchao Xiu

State Key Laboratory for Turbulence and Complex Systems, Department of Mechanics and Engineering Science, College of Engineering, Peking University, Beijing 100871, P. R. China
*Corresponding Author, E-mail: yy@pku.edu.cn

Abstract: In this paper, a long short-term memory network structure is developed to estimate state of charge for lithium-ion batteries owing to its time series characteristic. It is further followed by the extended Kalman filter to alleviate the process noise. The proposed algorithm shows reduced root mean squared error as low as 0.48%, compared with traditional algorithms like linear regression, support vector regression and general shallow neural network. Our work provides a feasible way to estimate state of charge of batteries for general dynamic loading conditions.

Key Words: Long short-term memory network, State of charge estimation, Extended Kalman filter, Lithium-ion batteries.

1 Introduction

Lithium-ion batteries are gaining in popularity since their first commercialization a few decays ago [1]. The outstanding characteristics like high energy density, light weight and longer lifespan [2] help them capture the headlines in not only portable electronic devices but also large products like electric vehicles (EVs) in particular. However, a precise estimation of state of charge (SOC), which is an essential part of the battery management systems (BMS), remains a challenging task. Just like the fuel gauge which indicates the amount of fuel in a fuel tank, SOC indicates the amount of charge left for the batteries in EVs. It provides critical information for EV drivers. SOC is a percentage value defined as the remaining charge divided by the full charge capacity, which is difficult to estimate because it is not a physical quantity to be directly measured. There are some basic ways to estimate SOC. For instance, the Coulomb counting [3] method is one of the classical estimation methodologies, where the charge loss is simply calculated by the integration of current through time. But it requires accurate current information from sensors and also needs knowledge about initial SOC value. In real applications, the open circuit voltage look-up table [4] is often used where the SOC is mapped to the open circuit voltage. It is simple and easy to implement and can be incorporated with temperature factors [5]. However, the open circuit voltage is hard to measure in real time since the batteries need to take a long-time rest before measuring. Besides, the open circuit voltage may respond differently among batteries and the method is not suitable for all types of batteries.

Various attempts have been exploited towards the model-based SOC estimation methods including electrochemical models and equivalent circuit models. A typical case of electrochemical models is the single particle model [6], which has the capability of describing not only the mass and energy transport of each component in a cell but also the current and voltage distributions. Besides this, the equivalent circuit models [7] use electrical circuit components like resistors, inductors and capacitors in an electrical network based on electrical equations.

On the other hand, data-driven methods [8] are catching peoples' eyes because they can deal with problems without specific knowledge of the physical models. Instead of establishing governing equations from physical laws, they give up all the equations and use only the data to perform the estimation. Under such circumstances, a relationship between the measured feature data and the target SOC value is created through a map like a linear regression (LR). However, a LR is too weak in many cases since such relationship may be nonlinear and very complicated. Support vector regression (SVR) adds more constraints based on LR to treat the nonlinearity. Nevertheless, the estimation function uses still just one equation which is not sufficient. Neural networks (NN) stand out for its effectiveness in nonlinear map where people can set multiple layers and multiple neurons in each layer of a network. Within this context, one can map input test features directly to output of SOC by designing artificial networks. For instance, He et al. [9] used artificial neural network based on measured current and voltage together with unscented Kalman filter to estimate SOC. But the method intrinsically does not consider the time series relationship. So, to take account of the previous times steps, a search algorithm was manually conducted and the estimation root mean squared error (RMSE) was within 2.5%. A long short-term memory (LSTM) network is a better candidate here because it adds time series connection in the structure compared to NN. Further, Chemali et al. [10] proposed a LSTM network structure to estimate SOC at changing temperatures. Similarly, Song et al. [11] combined convolutional neural network (CNN) and LSTM network in their work, and the achieved RMSE was within 2%. All these results highlight the viability of conducting various networks to estimate SOC by its powerful capability to map the nonlinearity from input features to output estimation. However, the estimation results are still relatively high where the parameters and network structures are probably not well optimized. In addition, different data sets are used in different studies and the comparison of estimation results among them is somewhat

*This work is supported by National Natural Science Foundation (NNSF) of China under Grants 61633001 and U1713223.

awkward because the estimation results may depend on the data in specific cases.

In this work, a LSTM network together with an extended Kalman filter (EKF) algorithm is proposed to estimate SOC. LSTM network captures the relationship from measured current, voltage, derivative of voltage and temperature to SOC using network connections. The EKF functions as a completing stage to remove noise after the LSTM estimation. Main results include:

- The prediction noise degrades after the EKF has been applied.
- The estimation errors show little variance for different test profiles which further supports the proposed model in general conditions.
- Linear regression, support vector regression and neural network are also conducted on the same data set for comparison and our proposed LSTM with EKF method shows the best performance.

The rest of the paper is organized as follows: Section 2 first introduces the test data set and then illustrates basic theory of LSTM, followed by the training process of the proposed network. Section 3 at first exhibits the LSTM prediction results and further discusses the comparison with other algorithms. Furthermore, the results after EKF for noise reduction are given. The paper is ended by conclusion in Section 4.

2 LSTM network

2.1 Battery test data

The battery test data are all acquired from online open source contributed by CALCE group [12]. 18650 lithium nickel manganese cobalt oxide battery cells with a rated capacity of 2000 mAh were used in the experiment. The cell has a diameter of 18 mm and is 65 mm in length with charge and discharge cut-off voltage at 4.2 V and 2.5 V, respectively, which is widely used in EVs in packs. The cells were put in a temperature chamber with adjustable physical property and outside connected to Arbin BT2000 battery test system to get charged or discharged with desired profiles.

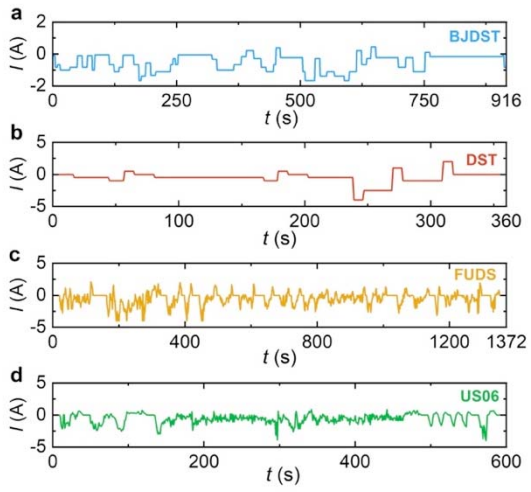

Fig. 1: The current regimes of 4 different EV testing profiles: (a) BJDST (b) DST (c) FUDS (d) US06. One test cycle is shown for each profile.

To mimic the working conditions of batteries in an EV in real applications, four commonly used testing profiles were adopted in the test including Beijing Dynamic Stress Test (BJDST) [13], dynamic stress test (DST), Federal Urban Driving Schedule (FUDS) [14] and the US06 driving schedule [15]. Fig. 1 shows the evolution of current I versus time in one cycle for each profile. The four test profiles are different in cycle time and have various structures where the FUDS and US06 curves seem more complicated. Current amplitude can be adjusted in proportion to the peak value for each profile according to users' will. Negative current values represent the discharging process. Note that there are still some positive values in each profile indicating the charging process, which is attributed to the fact that the vehicle will convert energy to the battery during braking.

During the test, each cell was first fully charged using the two constant-current/constant-voltage stages and then discharged to 80% SOC and rest for a while. From there, the cell was discharged according to one specific dynamic profile in Fig. 1 repeatedly until the voltage reached the discharge cut-off voltage at 2.5 V indicating 0% SOC. For each profile, cells were tested at 0, 25 and 45 ℃ respectively. A total of 12 sets of data were collected for 4 test profiles at 3 temperatures with each as a time series sequence. The SOC is used as the output of network described in the next section, while the input is the current I, voltage V, derivative of voltage dV and temperature T extracted directly from the system software recordings.

2.2 Theory of LSTM network

Considering the time series characteristic of the data where SOC evolves with time, the LSTM [16] network was utilized in the study. It is especially suitable for time series prediction. Moreover, it eliminates the need to manually choose previous time steps in the input data [9]. The LSTM layer differs from general neural network in that it will not only get information from the input layer but also pass information horizontally in time series by means of taking account of the previous hidden state and cell state.

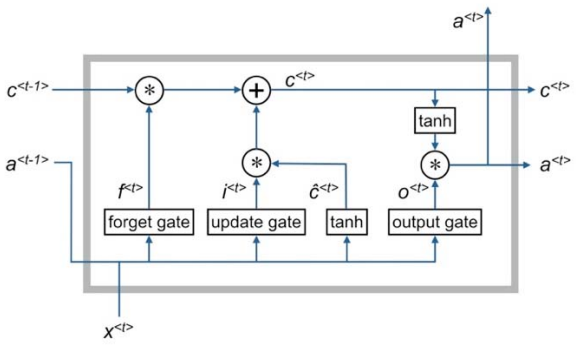

Fig. 2: Schematic of a LSTM cell. $f^{<t>}$, $i^{<t>}$ and $o^{<t>}$ are the forget gate, update gate and output gate, respectively, which are computed from the input vector $x^{<t>}$ and the previous hidden state $a^{<t-1>}$. The cell state $c^{<t>}$, together with the hidden state $a^{<t>}$, carries information horizontally to the next time step. Superscript t denotes the time step in the sequence.

Fig. 2 demonstrates how the LSTM cell functions. The LSTM cell memorizes information by using 'gates'. The forget gate $f^{<t>}$ and the update gate (also sometimes called

input gate) $i^{<t>}$ choose to memorize or forget information by an activation function with output between 0 and 1. Then the output gate $o^{<t>}$ controls the output of the current hidden state $a^{<t>}$. The hidden state a and the cell state c will pass along the time series. Only the hidden state a will transport upwards to the next hidden layer. The detailed functions are as follows:

$$\begin{aligned}
\hat{c}^{<t>} &= \tanh(W_{ca} * a^{<t-1>} + W_{cx} * x^{<t>} + b_c),\\
i^{<t>} &= \sigma(W_{ua} * a^{<t-1>} + W_{ux} * x^{<t>} + b_u),\\
f^{<t>} &= \sigma(W_{fa} * a^{<t-1>} + W_{fx} * x^{<t>} + b_f),\\
o^{<t>} &= \sigma(W_{oa} * a^{<t-1>} + W_{ox} * x^{<t>} + b_o),\\
c^{<t>} &= i^{<t>} * \hat{c}^{<t>} + f^{<t>} * c^{<t-1>},\\
a^{<t>} &= o^{<t>} * \tanh c^{<t>}.
\end{aligned} \quad (1)$$

Here $\hat{c}^{<t>}$ is the cell state candidate to be updated. All the weight matrices W and bias terms b are parameters to be learned during the training process and σ denotes the sigmoid activation function:

$$\sigma(z) = \frac{1}{1+e^{-z}}. \quad (2)$$

The superscript t or t-1 in angle brackets denotes the current time step or the previous time step.

2.3 Network structure and training

Fig. 3 illustrates the overall structure of the LSTM network. This is a 2-hidden layer network. Each input sample was a series of data so our data were first cut into small sequences. The time interval was 1 s and each sequence had a length of k = 600 time steps which corresponded to 10 minutes. A total of 314 sequence samples were derived. At every time step, the input was a vector $x^{<t>}$ including the current I, voltage V, derivative of voltage dV and the temperature T. All the samples were randomly arranged and then divided into 3 parts with 75% as training data, 15% as cross validation data and the rest 10% as test data. The implementation of the LSTM network is illustrated in Algorithm 1.

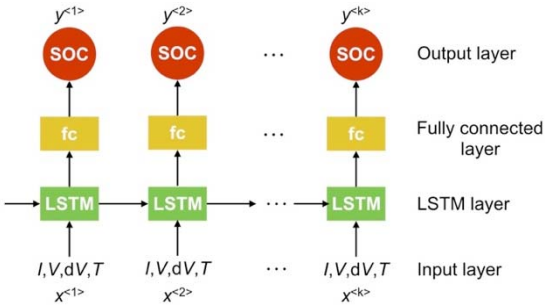

Fig. 3: Structure of the LSTM network. Each training sample contains a sequence of data of k time steps. The LSTM layer functions as the first hidden layer which passes information from the previous time step to the current step. Another fully connected layer is applied as the second hidden layer between the LSTM layer and the output SOC layer.

To minimize the loss function in step 5, the ADAM [17] optimization method was used which is a modification of gradient descent algorithm. The model was trained in mini-batch with size of 64. A total of 800 epochs were trained. The training process was taken in MATLAB with the deep learning toolbox. During training, cross validation data were used to compute the error between the actual value and the predicted value using the parameters from training results, since using training data alone lacks knowledge on other data and may cause overfitting. As a result, the model parameters were taken when the error for cross validation data reached a minimum. The model was then further tested on the test data set.

Algorithm 1 LSTM network implementation

1. Initialize training data

$$x = \frac{x-\mu}{\sigma},$$

where μ and σ are the mean and the standard deviation of x, respectively.

2. Compute hidden state $a^{[2]<t>}$ in the second LSTM layer according to (1), where superscript 2 in square bracket denotes the second layer in the network.

3. Compute hidden state $a^{[3]<t>}$ in the third fully connected layer by a weight matrix W and a bias vector b.

$$a^{[3]<t>} = W_a^{[3]} * a^{[2]<t>} + b_y^{[3]}.$$

4. Compute prediction $y^{<t>}$ in the fourth output layer

$$y^{<t>} = W_a^{[4]} * a^{[3]<t>} + b_y^{[4]}.$$

5. Compute and minimize the loss function for all time steps in all sequence samples

$$J = \frac{1}{2k}\sum_i (y^{<i>} - SOC^*)^2 + \lambda * \frac{1}{2} w^T w,$$

including a regularization term for weight elements w with factor λ to prevent overfitting. Here SOC^* is the actual value.

The major hyper parameters were learning rate α, regularization factor λ, number of hidden layer units and the sample sequence length. A range of different values were tested to find the optimal ones. Finally, the learning rate α was chosen as 0.1 and the number of units in the LSTM layer and fully connected layer were both chosen as 20. The regularization factor λ was chosen as 0.1. The sequence length which means the time steps in a sequence sample did not affect the results much and was chosen as 600. On the whole, hyper parameter tuning and the design of the network structure need many trials. In fact, initially a simpler 3-layer network with an input layer, a LSTM layer and an output layer was used, but prediction results showed large bias from the actual SOC during the first several minutes. Then a second hidden layer (fully connected layer) was added between the LSTM layer and the output layer as above and the prediction results converged almost with no delay. The network prediction results will be discussed in the following section.

3 Results and discussion

3.1 LSTM estimation results

After the LSTM network had been trained with all hyper parameters optimized, the model was applied to predict SOC for all of the data from 4 test profiles at 3 temperatures. The input is a time series of current I, voltage

V, derivative of voltage dV and temperature T. Using the structure shown in Fig. 3 with parameters already obtained including the weight matrices and bias terms in the training process, estimations of SOC in time series were predicted.

Fig. 4 shows two prediction examples of the DST and FUDS test profile both at 0 ℃. The batteries were discharged from 80% SOC with the DST (Fig. 4a) or the FUDS (Fig. 4b) profile and were depleted in about 10000 seconds. At first glance, the prediction lines seem overlap with the actual lines. No delay is observed in the beginning several minutes. Moreover, the LSTM predictions (in red line) show some small disturbance which is treated as measurement noise and will be addressed with a filtering process later. The detailed prediction error information is listed in Table 1.

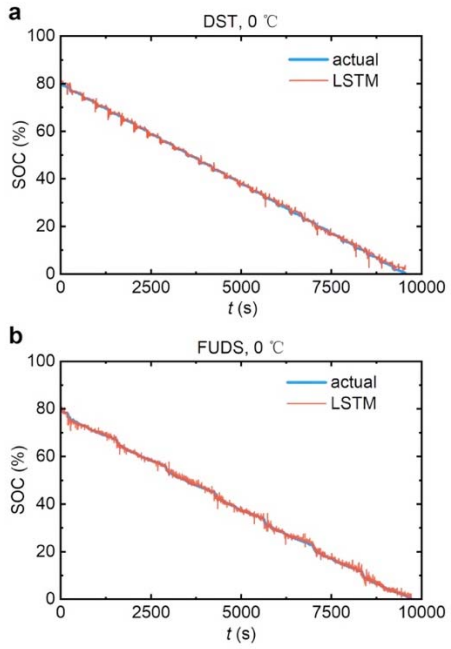

Fig. 4: LSTM network prediction results of SOC from 80% to 0% for (a) DST profile at 0 ℃ (b) FUDS profile at 0 ℃. The actual value is shown in the blue line for reference.

Table 1: The RMSE of the LSTM network predictions for different test profiles at different temperatures

RMSE (%) Test profile	Temperature (℃)		
	0	25	45
BJDST	0.71	0.34	0.26
DST	0.72	0.45	0.46
FUDS	0.68	0.37	0.39
US06	0.74	0.38	0.29

The RMSE adopted to measure the fitness of prediction data was computed as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{1}^{N}(prediction\text{-}actual)^2}\ . \qquad (3)$$

As can be seen from the table, for a fixed temperature the RMSE varies slightly between different test profiles, which means that our model is almost independent of loading conditions. It indicates that the network structure can fit general real-world dynamic currents. On the other side, however, there is still some variance for a fixed profile at different temperatures. The higher temperature seems to have more accurate predictions. For example, the error of tests at 0 ℃ is around 0.7, which is 2-3 times of those at 45 ℃. This is consistent with other literature [10] using machine learning algorithms to estimate SOC. In general, the overall error is below 1% using the LSTM network.

3.2 Comparison with other algorithms

To compare the proposed LSTM network with other machine learning algorithms, we conducted LR, SVR and NN on the same data set. These three algorithms do not consider the time series characteristic in LSTM and each training sample is just one time step in the sequence. The input of a sample is a vector containing features of the current I, voltage V, derivative of voltage dV and temperature T and the output is the SOC.

Since linear regression sometimes can be more efficient considering polynomial terms, we tried different power terms of input elements and found that higher power terms were more accurate. Given the computing cost, polynomial terms up to the 9^{th} power were adopted.

The type of SVR used in the work was ε-SVR [18]. The implementation of the algorithm was realized in MATLAB with the aid of libsvm [18] software.

The third algorithm used for comparison was NN. Unlike LSTM, the NN does not use connections in time series and samples are independent of each other in order. To make a comparison, a 2-hidden layer NN was developed in our work. The structure was similar to that of LSTM network but had no memory cells.

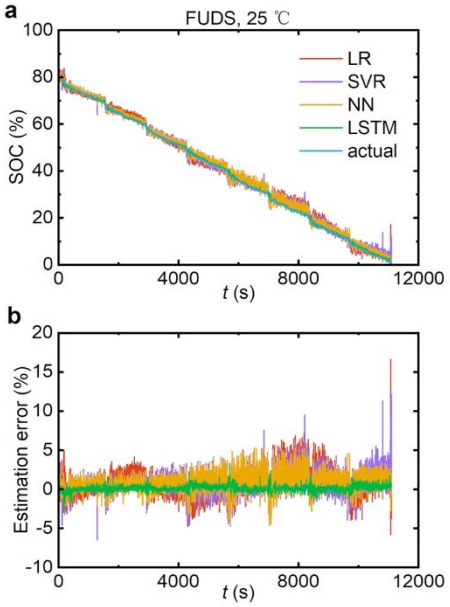

Fig. 5: (a) Comparison of LR, SVR, NN and LSTM estimation results for the FUDS testing profile at 25 ℃. (b) The estimation errors versus time in (a). The estimation errors have same colors as (a) and therefore are not labeled.

All the three algorithms were carefully tuned. Fig. 5 exhibits the comparison of estimation results using the different algorithms in one example under FUDS testing profile at 25 ℃. Detailed error results are shown in Table 2. Since errors are not sensitive to testing profiles, they are listed by temperature. In general, the LR has the largest estimation error and it reaches 2.71% at 0 ℃. Higher temperature at 45 ℃ has a smaller error at 1.55%. LR performs worst among the four algorithms since it is a bare linear combination for prediction. But the structure is simple and the parameters are easy to tune. For the SVR, it can be treated as an improvement of linear regression by means of introducing slack variables [18]. The errors are a little bit smaller than those of LR. For instance, the RMSE changes from 2.71% to 2.43% at 0 ℃ but is still high. The third column shows the NN errors which are smaller than SVR and but larger than LSTM. Neural networks use neurons in multiple layers which can be viewed as a bunch of linear combinations, so the performance is better than LR or SVR. However, the flexibility of structure including the choice of layers and neurons in each layer makes the tuning process complicated. Moreover, it does not consider the inner time series connections among test steps. LSTM network proves to be the best candidate in our example. RMSE is only 0.36% at 45 ℃. The average error is below 0.8% which is approximately half of NN, although there are some fluctuations at different temperatures. One difficulty of implementing LSTM network is the hyper parameter tuning process including the design of input sequence data. Besides, the structure of the network needs to be widely tried as in our example adding the second hidden layer solves the problem of large bias in the beginning. On the whole, the LSTM network performs better than other three algorithms attributed to its complicated combinations of neurons and also the time series connection by gates.

Table 2: Comparison of SOC estimation error among LR, SVR, NN and LSTM at 0, 25 and 45 °C

Temperature (℃)	RMSE			
	LR (%)	SVR (%)	NN (%)	LSTM (%)
0	2.71	2.43	2.01	0.72
25	1.84	1.21	1.06	0.39
45	1.55	0.71	0.53	0.36

3.3 Improvement by EKF

Finally, in order to smooth the sharp edges of noise in the prediction lines in Fig. 4, EKF was applied on the LSTM predictions. EKF provides an algorithm to give an estimate of unknown variables, which has been broadly used in navigation or object tracking fields. The algorithm is based on two functions:

State transition function:

$$SOC(k) = SOC(k\text{-}1) - \frac{I^{*}\Delta t}{Capacity} * 100\% + v,$$

$$v \sim \mathcal{N}(0, Q)$$

Measurement function:

$$LSTM(k) = SOC(k) + w, w \sim \mathcal{N}(0, R)\,. \qquad (4)$$

Both the state transition function and the measurement function have noise terms assumed to be zero-mean Gaussian with variance Q and R, respectively. The filter algorithm gives the best prediction of the variable $SOC(k)$ by multiplying the two noise terms in an iterative way [19]. Based on our knowledge, the state variance of initial SOC was set as 0.2, and the variance terms Q and R were set as 0.01 and 0.2, respectively. The algorithm was performed on MATLAB with the aid of control system toolbox.

Fig. 6 displays the filtering results for DST and FUDS profiles at 0 ℃. Compared with Fig. 4 without filtering, the LSTM predictions with EKF become smoother. The sharp edges of noise are reduced. A detailed error table after the filter algorithm is shown in Table 3. Each error term is the average for four test profiles at three temperatures in one algorithm. The EKF works for all algorithms and the error reduced within 0.2% in general for all these algorithms. The LSTM prediction result after the EKF algorithm is as low as 0.48% while other algorithms are beyond 1%. Our prediction errors are reduced than using a single LSTM network [10], though different experiments are conducted there. The proposed LSTM network structure together with an EKF gains the best result among all methods.

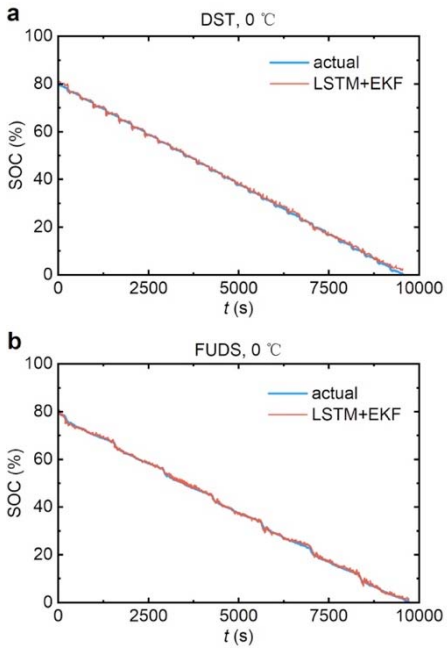

Fig. 6: Filtering results of SOC after EKF is applied on LSTM prediction for (a) DST profile at 0 ℃ (b) FUDS profile at 0 ℃. The actual value is shown in the blue line for reference.

Table 3: Comparison of SOC estimation error with EKF for different algorithms

Algorithm	RMSE (%)
LR	2.06
SVR	1.58
NN	1.31
LSTM	0.54
LSTM+EKF	0.48

4 Conclusion

In this paper, the SOC of lithium-ion batteries is estimated by LSTM network and EKF. The LSTM network chooses input data including current, voltage, derivative of voltage and temperature and predicts SOC as output. A further EKF is adopted to flatten the noise in prediction. The proposed model is compared with linear regression, support vector regression and neural network on the same data. The average RMSE is 0.48% for our algorithm which is less than half of others. Besides, the LSTM network eliminates the need of manually choosing previous time steps in neural network. Comparison tests on the same data set also strengthen our confidence of the superiority of the proposed method. Finally, prediction is almost independent of test profiles, indicating that the method can be applied to more general working conditions.

References

[1] Nishi, Y., The development of lithium ion secondary batteries. *The Chemical Record*, 2001, 1(5), 406-413.

[2] Taracson, J.; Armand, M., Issues and challenges facing lithium ion batteries. *Nature*, 2001, 414, 359-367.

[3] Ng, K. S.; 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), 1506-1511.

[4] Weng, C.; Sun, J.; Peng, H., A unified open-circuit-voltage model of lithium-ion batteries for state-of-charge estimation and state-of-health monitoring. *J. Power Sources*, 2014, 258, 228-237.

[5] 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.

[6] Tanim, T. R.; Rahn, C. D.; Wang, C. Y., State of charge estimation of a lithium ion cell based on a temperature dependent and electrolyte enhanced single particle model. *Energy*, 2015, 80, 731-739.

[7] He, H.; Xiong, R.; Fan, J., Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *Energies*, 2011, 4(4), 582-598.

[8] Xiong, R.; Cao, J.; Yu, Q.; He, H.; Sun, F., Critical review on the battery state of charge estimation methods for electric vehicles. *IEEE Access*, 2017, 6, 1832-1843.

[9] He, W.; Williard, N.; Chen, C.; Pecht, M., 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.*, 2014, 62, 783-791.

[10] 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.*, 2017, 65(8), 6730-6739.

[11] 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, 88894-88902.

[12] Zheng, F.; Xing, Y.; Jiang, J.; Sun, B.; Kim, J.; Pecht, M., Influence of different open circuit voltage tests on state of charge online estimation for lithium-ion batteries. *Appl. Energy*, 2016, 183, 513-525.

[13] Ma, Z.; Jiang, J.; Shi, W.; Zhang, W.; Mi, C. C., Investigation of path dependence in commercial lithium-ion cells for pure electric bus applications: Aging mechanism identification. *J. Power Sources*, 2015, 274, 29-40.

[14] Hunt, G., USABC electric vehicle battery test procedures manual. *Washington, DC, USA: United States Department of Energy*, 1996.

[15] www.epa.gov/sites/production/files/2015-10/us06col.txt. (accessed on 29 February 2020)

[16] Hochreiter, S.; Schmidhuber, J., Long short-term memory, *Neural Comput.*, 1997, 9(8), 1735-1780.

[17] Kingma, D. P.; Ba, J., Adam: A method for stochastic optimization. *arXiv preprint arXiv*:1412.6980 2014.

[18] Chang, C. C.; Lin, C. J., LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2011, 2(3), 1-27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[19] Kim, Y.; Bang, H., Introduction to Kalman Filter and Its Applications. In *Introduction and Implementations of the Kalman Filter*. IntechOpen, 2018.