

# Lithium-ion Battery State of Charge Estimation Method Using Optimized Deep Recurrent Neural Network Algorithm

M.S. Hossain Lipu<sup>a</sup>, *Student Member, IEEE*, M. A. Hannan<sup>b\*</sup>, *Senior Member, IEEE*, Aini Hussain<sup>a</sup>, M.H.M. Saad<sup>a</sup>, A. Ayob<sup>a</sup>, K. M. Muttaqi<sup>c</sup>, *Senior Member, IEEE*

<sup>a</sup>Centre for Integrated Systems Engineering and Advanced Technologies, FKAB, Universiti Kebangsaan Malaysia, 43600 Bangi, Malaysia.

<sup>b</sup>Department of Electrical Power Engineering, Universiti Tenaga Nasional, Kajang, 43000, Selangor, Malaysia

<sup>c</sup>School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, NSW, Australia

\*Corresponding author: hannan@uniten.edu.my (M A Hannan)

**Abstract**— This paper presents an enhanced machine learning based state of charge (SOC) estimation method for a lithium-ion battery using a deep recurrent neural network (DRNN) algorithm. DRNN is suitable for SOC evaluation due to strong computation intelligence and self-learning capabilities. Nevertheless, the performance of DRNN is constrained due to the training accuracy and duration which entirely depends on the appropriate selection of hyper-parameters including hidden layer and hidden neurons. Therefore, firefly algorithm (FA) is employed to find the optimal number for hyper-parameters of DRNN networks. The optimized DRNN based FA algorithm for SOC estimation does not require extensive knowledge about battery chemistry, electrochemical battery model and added filter, rather only needs battery test bench to measure current and voltage. The developed model is tested using two different types of lithium-ion batteries namely lithium nickel manganese cobalt oxide (LiNiMnCoO<sub>2</sub>) and lithium nickel cobalt aluminum oxide (LiNiCoAlO<sub>2</sub>). The proposed model is validated by two experimental tests; one with static discharge test and other with pulse discharge test at room temperature. The experimental results indicate the superiority of the DRNN based FA method in comparison with the back-propagation neural network (BPNN) and radial basis function neural network (RBFNN).

**Index Terms**— State of charge, Lithium-ion battery, Deep recurrent neural network, Levenberg-Marquardt algorithm, Firefly algorithm

## I. INTRODUCTION

Carbon emission has significant measurable effects on the environment such as temperature rise, ice melting, extreme weather events like hurricanes and lightning. The applications of energy storage technologies have received massive attention due to their huge contribution in reducing carbon emissions. The lithium-ion battery has the lucrative characteristics of fast charging, high voltage, high energy density, and long life cycle and hence it is widely used in the automotive industry [1]. However, lithium-ion battery is still suffering from some problems such as power electronics controller interface, accurate charge estimation, temperature

control, power management, cost, and safety concern [2]. Hence, the advanced researches are concerned greatly in evaluating the charge as well as in controlling and converting power effectively in electric vehicle (EV) applications [3].

Battery management system (BMS) does the necessary operations to run EV properly such as accurate state of charge (SOC), state of health (SOH) estimation, efficient control of charging and over discharging, thermal management, safety, and protection. SOC is a vital component in BMS which is defined as the remaining charge presented inside a battery cell. SOC is highly explored research topic to evaluate the lithium-ion battery performance. Accurate estimation of SOC is of great significance to prolong battery life time as well as to protect the battery from being overcharged. Nevertheless, SOC is an internal parameter of a lithium-ion battery which not only depends on battery chemistry, chemical reactions but also on numerous issues such as aging, temperatures. Hence, further exploration is required to develop an advanced SOC estimation algorithm under various uncertainties.

## A. Related Works

SOC estimation methods can be categorized into three groups; conventional method, model-based method, and machine learning method. Open circuit voltage (OCV) [4] and coulomb counting (CC) [5] techniques are the conventional estimation approaches which use voltage and discharge current directly to estimate SOC. Nevertheless, the online operation cannot be executed using OCV method. CC suffers from the accumulation of measurement error due to the current integration. In order to overcome the above challenges, model-based SOC estimation methods have been introduced. Kalman Filter (KF) [6], Particle filter (PF) [7] and H<sub>∞</sub> Filter [8] are the commonly used model-based approaches and have become popular due to the strong capability to handle noises in the measured values. Nonetheless, KF does not deliver satisfactory results in a highly nonlinear system. Moreover, the KF accuracy is highly depends on battery model parameters. Particle filter (PF) based SOC estimation provides high accuracy with fast estimation speed. Nevertheless, the PF method requires a complex mathematical tool. H<sub>∞</sub> Filter based SOC estimation has reasonable accuracy and fast computational cost. However, the accuracy of H<sub>∞</sub> Filter could

diverge due to the aging, hysteresis and temperature effects. Machine learning methods such as artificial neural network (ANN) [9], fuzzy logic (FL) [10] and support vector machine (SVM) [11] have received huge interest in SOC estimation in recent years due to their strong computation intelligence capabilities. Artificial neural network (ANN) is a popular subclass of machine learning method that can examine SOC accurately considering lithium-ion battery non-linear characteristics, aging, noises, and temperature effects. Nonetheless, ANN needs a vast amount of data, storage device and has a long training duration. SVM has accurate and fast estimation but suffers from the high complex computation.

### B. Major Contributions

In this research, we propose an improved machine learning algorithm for SOC estimation using a deep recurrent neural network (DRNN) algorithm. In particular, an optimized DRNN based SOC estimation model is developed using Firefly algorithm (FA). The major contributions of this work are outlined as follows:

- A new DRNN based FA model has been developed which can evaluate SOC accurately and directly by measuring signals from the battery such as current, voltage and temperature, hence avoiding added filter used in the model-based approaches.
- The DDRN algorithm can self-update its own network parameters including weights and bias as well as use the previous and present information of hidden state to estimate SOC. This is a completely different from electrochemical battery model based SOC estimation which needs extensive knowledge and time about model parameter estimation.
- The conventional DRNN based SOC estimation model uses exhaustive trial and error approach to search for the hidden layers and hidden neurons. However, data under-fitting and over-fitting problems make the DRNN algorithm unsuitable for SOC estimation. Therefore, FA is employed to find the optimal hyper-parameters of DRNN algorithm which helps to assess SOC accurately under different charge and discharge cycles.
- The generalization of DDRN based FA method for SOC estimation is validated under different lithium-ion battery chemistries. Though the DDRN takes substantial time for training operation, the SOC can be measured very quickly due to low computational complexity in the testing stage and has the advantage to perform SOC evaluation without disconnecting the battery from the load.

## II. SOC ALGORITHM FRAMEWORK

### A. Deep Recurrent Neural Network Algorithm

The deep recurrent neural network (DRNN) is particularly suitable for its powerful tool to address time series problems [12]. The DRNN is successfully implemented for parameter projecting in numerous application such as industries, image processing, and forecasting [13]. Moreover, DRNN comprises a unique dynamic memory, through which complex system can be addressed with the appropriate value of weights. The

learning procedure of the DRNN is implemented through one of the two ways such as feed-forward connection and feedback connection [14]. Although the training process of DRNN has some similarity with the feed forward neural network, but there are some differences between the two processes. The output response is evaluated based on a repeated feedback process which contains the hidden output of that instance and hidden output from the previous instance. The information is stored on the feedback loop of the previous phase and final output is predicted based on the instantaneous output and the previous output [15]. The basic structure of DRNN is presented in Fig. 1.

The SOC is computed using the DRNN algorithm at time  $t$  with input series ( $x = x_1 \dots x_i$ ), hidden vector series ( $h = h_1, \dots, h_i$ ) and output vector  $y_k$ . The mathematical expressions of the series are shows below.

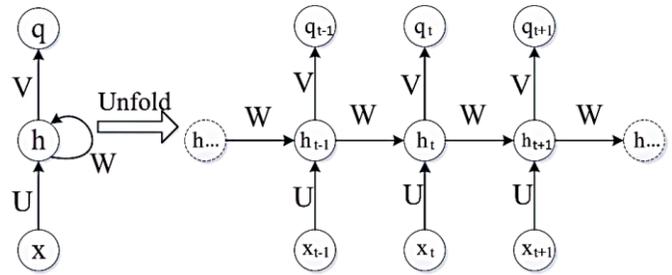


Fig. 1 Structure of DRNN

$$net_j = \sum_j w_{i,j}x_i + w_{hh}h_{i-1} + \theta_{i,j} \quad (1)$$

$$O_j = f(net_j) \quad (2)$$

$$net_k = \sum_k W_{j,k}O_j + \theta_{j,k} \quad (3)$$

$$O_k = f(net_k) \quad (4)$$

Where  $x_i$  denotes the weight between input layer and hidden layer,  $w_{hh}$  denotes the weight between a hidden layer and itself at adjacent time steps,  $x_j$  is the weight between hidden layer and output layer.  $O_j$  and  $O_k$  presents the output of hidden layer and output layer respectively. The hidden layer bias and output layer bias are characterized by  $\theta_{j,i}$  and  $\theta_{j,k}$ .  $f()$  denotes the sigmoid activation function which is defined as

$$f(net) = \frac{1}{1+e^{(-net)}} \quad (5)$$

However, two training algorithms can be used to train the DRNN such as back-propagation through time (BPTT) and other is real-time recurrent learning (RTRL). In particular, the network parameter is changed in BPTT from feedback to feed-forward structures. For the purpose of this research, the BPTT method is implemented, which contains two major stages namely forward pass and backward pass [16]. The output of the forward pass stage is assessed utilizing the inputs, weight, bias and sigmoid activation function. In contrast, the backward pass algorithm estimates the error and propagates from the output layer to the hidden layer [17]. The error in the output layer is estimated through the mathematical expression below.

$$e_k = T_k - O_k \quad (6)$$

Where,  $T_k$  denotes the actual output.

The error depends on the factor ( $\partial_k$ ) that distributes the output error ( $T_k$ ) and executes the data upgradation for previous layers as expressed in the following equation,

$$\partial_k = e_k f'(net_k) \quad (7)$$

The factor  $\partial_j$  is computed in the hidden layer which depends on the derivative of its activation function and the error factor in the output layer ( $\partial_k$ ) as presented in the following equation,

$$\partial_j = f'(net_j) \partial_k w_{j,k} \quad (8)$$

Weights are updated using the following equations,

$$\Delta w_{j,k} = \alpha \partial_k \partial_j \quad (9)$$

$$w_{j,k} = w_{j,k} + \Delta w_{j,k} \quad (10)$$

$$\Delta w_{hh} = \alpha \partial_j h_{i-1} \quad (11)$$

$$w_{hh} = w_{hh} + \Delta w_{hh} \quad (12)$$

$$\Delta w_{i,j} = \alpha \partial_j x_i \quad (13)$$

$$w_{i,j} = w_{i,j} + \Delta w_{i,j} \quad (14)$$

where  $\alpha$  denotes the learning rate.

Biases are updated using the following equations

$$\Delta \theta_{j,k} = \alpha \partial_k \quad (15)$$

$$\theta_{j,k} = \theta_{j,k} + \Delta \theta_{j,k} \quad (16)$$

$$\Delta \theta_{i,j} = \alpha \partial_j \quad (17)$$

$$\theta_{i,j} = \theta_{i,j} + \Delta \theta_{i,j} \quad (18)$$

The Fig. 2 depicts the pseudo code of BPTT algorithm.

BPTT Algorithm	
1:	<b>Procedure:</b> TRAIN
2:	<b>X:</b> ← Training Data Set of Size mxm
3:	<b>y:</b> ← Labels for records in X
4:	<b>w:</b> ← The weights for respective layers
5:	<b>l:</b> ← The of layers in neural network, 1 ... L
6:	$D_{ij}^{(l)} \leftarrow$ the error
7:	$t_{ij}^{(l)} \leftarrow 0$ . For all $l, i, j$
8:	<b>For</b> $i = 1$ to $m$
9:	$a^1 \leftarrow$ feedforward ( $a^{(l)}, w$ )
10:	$d^l \leftarrow a(L) - y(i)$
11:	$t_{ij}^{(l)} \leftarrow t_{ij}^{(l)} + a_i^{(l)} \cdot t_{ij}^{(l+1)}$
12:	<b>If</b> $j \neq 0$ then
13:	$D_{ij}^{(l)} \leftarrow \frac{1}{m} t_{ij}^{(l)} + w_{ij}^{(l)} \cdot \lambda$
14:	<b>else</b>
15:	$D_{ij}^{(l)} \leftarrow \frac{1}{m} t_{ij}^{(l)}$
16:	where $\frac{\partial}{\partial w_{ij}^{(l)}} J(w) = D_{ij}^{(l)}$

Fig. 2 The pseudo code of BPTT algorithm [18]

### B. Levenberg-Marquardt Algorithm

The Levenberg-Marquardt (LM) algorithm is known to be the fastest training algorithm which is developed based on the approximation of Newton method [19], [20]. The back-propagation algorithm uses the first order error approximation for the neural network computation [21]. The weight of the DRNN is updated by LM through the following expression.

$$\Delta w = [\mu I + \sum_{p=1}^P J^P(w)^T J^P(w)]^{-1} \nabla E(w) \quad (19)$$

where,  $J^P(w)$  denotes the Jacobian matrix of the error vector  $e^P(w)$  which is computed in  $w$ ;  $I$  denotes the identity matrix;

The vector error represents the error of the network pattern denoted as  $P$ , where  $P$  is computed as

$$e^P(w) = t^P - c^P(w) \quad (20)$$

Here, the parameter  $\mu$  in the equation is either increased or decreased in each step of the computation. In this procedure, the error value  $\mu$  is divided by a factor of  $\beta$ , if the error value is reduced.

The steps of the LM algorithm are executed through the process shown in Fig. 3. In this method, the network output, the error vectors and Jacobian matrix of each pattern are computed. After that, the error is also recalculated via  $w + \Delta w$  as network weight. However, if the value of the error is reduced then  $\mu$  is divided by a factor of  $\beta$  and the new found weight is kept before the process starts again. In other cases, the error value is multiplied with the factor of  $\beta$ , then the change of error  $\Delta w$  is again computed and iteration process continues.

LM Algorithm	
<b>while</b> not stop-Criterion <b>do</b>	
Calculate $C^P(W)$ for each pattern	
Repeat	
$e_2 = \sum_{p=1}^P 1. e^P(W)^T \cdot e^P(W)$	
Calculate $J^P(w)$ for each pattern	
$e^P(w + \Delta w)^T \cdot e^P(w + \Delta w)$	
<b>if</b> $e_1 \leq e_2$ <b>then</b>	
$\mu = \mu * \beta$	
<b>End if</b>	
<b>Until</b> $e_1 < e_2$	
$\mu = \mu / \beta$	
$W = w + \Delta w$	
<b>end while</b>	

Fig. 3 The pseudo code of LM algorithm [22]

### C. Firefly algorithm

Firefly algorithm (FA) is based on the flashing light of fireflies in the summer sky [23]. The patterns of these flashes of light are unique for each species of the fireflies and the flashes are used for different purpose in different communicating such as attracting the potential prey as well as attracting the mating partners. This concept of flashing characteristics becomes the basis to develop the FA [24]. The FA is developed using three statements. The first statement says that all fireflies are unisex and hence the attraction between them is independent. The second statement outlines the attractive force between the fireflies which is proportional to their brightness, which means that the lesser bright firefly will be attracted by the brighter ones. The same brightness fireflies will move randomly within the boundary. The third statement defines the objective function which will be determined by the brightness of firefly [25], [26].

The FA has two vital components; one is the light intensity and other is the formulation of attractiveness. The attraction function,  $\beta(r)$  is defined from the relationship between firefly attraction and light intensity, as shown in the following equation.

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \quad (m \geq 1) \quad (21)$$

where  $\beta_0$  is the attractiveness for  $r = 0$ ,  $\gamma$  is the light absorption coefficient while  $r$  is the Cartesian distance between two fireflies as defined,

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (22)$$

where the two fireflies at  $x_i$  and  $x_j$  are characterized by  $i$  and  $j$ . The  $k$ -th component of the spatial coordinate  $x_i$  of the  $i$ -th firefly is denoted by  $x_{i,k}$ . The movement of the firefly  $i$  which feels attraction towards the brighter one (firefly  $j$ ) is expressed as,

$$x_{i-new} = x_{i-old} + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_{i-old}) + \alpha_r \left( rand - \frac{1}{2} \right) \quad (23)$$

where the second term in the above equation relates to the attraction and the third term is linked to the randomization. The randomization parameters  $\alpha_r$  is found in the third term. The function *rand* will generate random number consistently between '0' and '1'. There is a relationship between decreasing function  $\alpha_r$  and decreasing factor,  $\delta$ , as illustrated below:

$$\alpha_r(t+1) = \alpha_r(t) \times \delta \quad (24)$$

The pseudo code of FA is shown in Fig. 4

```

Firefly Algorithm
Begin
Objective function  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ 
Generate initial population of fireflies  $X_i (i = 1, 2, \dots, n)$ 
Light intensity to  $f(X_i)$ 
Define light absorption coefficient  $\gamma$ 
while ( $t < \text{MaxGeneration}$ )
for  $i = 1: n$  all  $n$  fireflies
for  $i = 1: n$  all  $n$  fireflies (inner loop)
if ( $I_j > I_i$ )
Move firefly  $i$  toward  $j$ 
end if
Attractiveness varies with distance  $r$  via  $e^{-\gamma r}$ 
Evaluate new solutions and updates light intensity
end for  $j$ 
end for  $i$ 
Rank the fireflies and find the current best
end while
Postprocess results and visualizations
End

```

Fig. 4 The pseudo code of Firefly algorithm [27]

### III. DRNN BASED FA DESIGN FOR SOC ESTIMATION

The FA is established using three important essentials, namely, input information, objective function, and optimization constraints. Each essential is working for enhancement and classification to achieve optimal hyper-parameters of DRNN. The aim of FA is to search for the appropriate value of hyper-parameters of DRNN in order to reach the best solution by minimizing the objective function while maintaining the optimization constraints in each generation during the iterative process.

#### A. Input Information

The input data for the FA optimization method is a number of boundary values for the input dataset of the hyper parameters of DRNN. The input matrix is developed using the number of columns and rows. The number of columns corresponds to the population of hyper parameter within the

boundary and the number of rows relates to the number of problem dimensions, as shown in the following matrix.

$$D_{ij} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1j} \\ X_{21} & X_{22} & X_{23} & \dots & X_{2j} \\ X_{31} & X_{32} & X_{33} & \dots & X_{3j} \\ X_{41} & X_{42} & X_{43} & \dots & X_{4j} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ X_{i1} & X_{i2} & X_{i3} & \dots & X_{ij} \end{bmatrix} \quad (25)$$

where  $D_{ij}$  is the input data matrix which is defined by  $i$  and  $j$ .  $i = 1, 2, \dots, P$ ,  $P$  is the number of population;  $j = 1, 2, \dots, N$ ,  $N$  is the problem dimension.

#### B. Objective Function

The target value of the FA optimization is defined by an objective function. The aim of objective function is to find the lowest error rates through an iterative process which not only provides the best value of hyper parameters of DRNN algorithms but also delivers an accurate SOC estimation. In this research, root means square error (RMSE) is chosen as the objective function due to high sample dataset and random distribution of SOC error estimation [28]. The RMSE function is calculated using the following equation,

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (SOC_{a_i} - SOC_{es_i})^2} \quad (26)$$

where the actual SOC and estimated SOC value are represented by  $SOC_a$  and  $SOC_{es}$  respectively and  $N$  represents the amount of data observations.

#### C. Optimization Constraints

The FA algorithm must follow the optimization constraints while searching for the optimal values and evaluating SOC. The minimum and maximum ranges of hyper-parameters of DRNN including hidden layer, and hidden neurons are fixed to define the minimum search space. The population of hyper parameters is checked repeatedly whether they out of the boundary region. Else, the FA optimization could diverge which in turn deliver poor SOC estimation results. For instance, the variable  $X_{i,j}^k$  should be between  $X_{i,j}^{k-1}$  and  $X_{i,j}^{k+1}$ . If the variable  $X_{i,j}^k$  is greater than  $X_{i,j}^{k+1}$  or less than  $X_{i,j}^{k-1}$ , the results will be updated and placed within the border. Hence, the appropriate border of the hyper-parameters of DRNN must satisfy the constraints mentioned below,

$$X_{i,j}^{k-1} < X_{i,j}^k < X_{i,j}^{k+1} \quad (27)$$

### IV. LITHIUM-ION BATTERY EXPERIMENTS AND DATA PREPARATION

#### A. Lithium-ion battery cell

The experiments were conducted using ICR18650-26F lithium-ion batteries developed by Samsung. This battery is designed using  $\text{LiNiMnCoO}_2$  (LiNMC) as cathode and graphite as an anode. The test battery has the nominal capacity and the nominal voltage of 2600 mAh and 3.7V, respectively. The maximum charging voltage and cut-off voltage is set to 4.2 V and 2.75 V. The battery is charged using constant current-constant voltage (CC-CV) method. Another lithium-ion battery cell NCR18650B is used in this research which is

made by Panasonic. This battery has rated capacity of 3200 mAh and is built using graphite anode and LiNiCoAlO<sub>2</sub> cathode (LiNCA) as a cathode. The nominal and cut-off voltage is 3.6 V and 2.5 V respectively. This battery is popular due to high energy and power densities, long life span. The detailed specifications of the two lithium-ion battery are shown in Table I [29] [30].

TABLE I  
LITHIUM-ION BATTERY SPECIFICATIONS

Parameters	LiNiMnCoO <sub>2</sub>	LiNiCoAlO <sub>2</sub>
Rated nominal capacity (Ah)	2.6 Ah	3.2 Ah
Nominal Voltage (V)	3.7	3.6
Min/Max voltage (V)	2.75/4.2	2.5/4.2
Specific Energy (Wh/kg)	150-220	200-260
Cycle life	1000-2000	500

### B. Battery experimental tests

Different types of battery tests are conducted to validate the SOC estimation accuracy and robustness such as the static discharge test and Hybrid Pulse Power Characterization (HPPC) test. The battery is completely charged before the experiments which means initial SOC is set to be 100%. LiNMC battery is used to describe the tests. Similar approaches can be applied for LiNCA battery while maintaining the defined voltage and current value suggested by the manufacturer.

1) *Static discharge test*: Static discharge test is widely used to verify the SOC estimation accuracy. In general, the test is conducted using constant current discharge after the battery is being fully charged. The stages of static discharge test are explained as follows:

- i. First, the battery is charged using the constant current (CC) method with 1.3 A (0.5 C) current until the charge voltage reaches 4.2 V.
- ii. After, the constant voltage (CV) method is applied with the voltage of 4.2 V until the charge current drops to 0.13 A (0.05 C).
- iii. Check whether the battery is fully charged. If the battery is fully charged then, step iv will start, otherwise, step ii will begin.
- iv. The battery is rested for 1 hour.
- v. The battery is discharged with 2.6 A (1 C) until the battery voltage declines to 2.75 V.
- vi. Check whether the lower cut off voltage of the battery is reached. If the battery reaches at 2.75 V, then test ends otherwise, step v will initiate again.

2) *HPPC test*: HPPC test is a combination of sequence charge and discharge pulses. The customized HPPC is employed in this research using the different rates of charge and discharge current rates. The steps of HPPC are described as follows:

- i. The battery is charged with constant current of 1.3 A (0.5 C) until the charge voltage reaches 4.2 V.
- ii. The battery is charged with the constant voltage of 4.2 V until the charge current drops 0.13 A (0.05 C).

- iii. Check whether the battery is fully charged. If the battery is fully charged then, step iv will start, otherwise, step ii will begin again.
- iv. The battery is rested for 1 hour.
- v. The battery is discharged at 1.3 A (0.5 C) for 10 seconds.
- vi. The battery is rested for 3 minutes.
- vii. The battery is charged at 1.3 A (0.5 C) for 10 seconds.
- viii. The battery is rested for 3 minutes.
- ix. The battery is discharged for 0.65 A (0.25 C) for 24 minutes to reduce the SOC capacity by 10%.
- x. Check whether the lower cut off voltage of the battery is reached. If the battery reaches 2.75 V, then test ends otherwise, step iii will continue.

### C. Battery Experimental Setup

A lithium-ion battery test bench model is developed which is divided into two parts, namely hardware part and software part, as displayed in Fig. 5. The hardware part comprises a LiNMC, LiNCA battery and NEWARE BTS-4000. The BTS-4000 is a fourth-generation battery testing system developed by NEWARE, and has been available in the market since 2008. Different kinds of battery tests can be performed using BTS-4000 including pulse test, resistance test, and cycle test. NEWARE BTS-4000 is better than other battery testing system in term of accuracy, data acquisition frequency and response time. The current and voltage range of NEWARE BTS-4000 is 6A and 5 V respectively. The precision of current and voltage accuracy is set as  $\pm 0.05\%$  FS [31]. It has eight independent channels which can record the battery data including current, voltage, charging and discharging capacity and cycle number. The device contains the power supply to charge the battery and load to discharge the battery and controller to control battery charging and discharging. The software part includes BTS software version 7.6 and MATLAB 2015a which are installed in the host computer. The static discharge test and HPPC test are conducted in each second at room temperature and data are stored in the host computer. The control operation of LiNMC and LiNCA battery charging and discharging is performed using NEWARE BTS-4000 while maintaining the threshold voltage and current values recommended by the manufacturer through BTS software version 7.6. Finally, the MATLAB 2015a version is used to estimate SOC and different error rates terms using DRNN-FA algorithm code and data obtained from experiments.

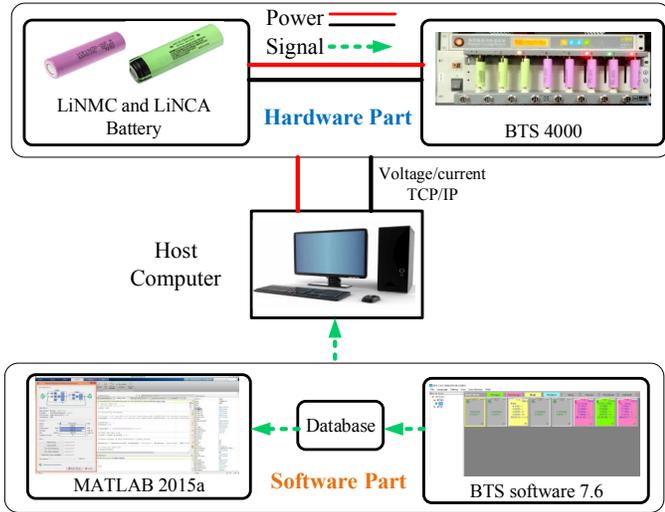


Fig. 5. Lithium-ion battery test bench model configuration

#### D. Training and Testing Dataset

After the algorithm development followed by the experimental data measurements, the whole dataset is divided into two subsets; training and testing. The developed SOC method is trained using 70% data and remaining unseen 30% data is used for SOC testing. The DRNN training and FA optimization execution for finding the optimal hyper-parameters are performed in offline condition while the SOC estimation is evaluated in online condition. Before the data training operation begins, data normalization is executed in order to enhance the convergence rate the DRNN algorithm. The boundary of data normalization is assigned to be  $[-1, 1]$  as expressed in the following equation,

$$x = \frac{2(x-x_{min})}{x_{max}-x_{min}} - 1 \quad (28)$$

Where the maximum and minimum value of input vector  $x$  is denoted by  $x_{max}$  and  $x_{min}$ . The maximum number of epochs during training the stage is set to be 1000. The performance goal is fixed to be 0.000001. The algorithm is executed on Core i5 2.3 GHz processor with 12 GB RAM. The input dataset including current and voltage of static discharge test and HPPC test for LiNMC and LiNCA battery are depicted in Fig. 6 and Fig. 7 respectively. The increment of voltage profile with respect to SOC for LiNMC and LiNCA battery is indicated in Fig. 8.

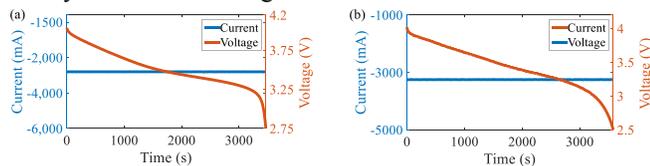


Fig. 6. Static discharge current profile (a) LiNMC (b) LiNCA

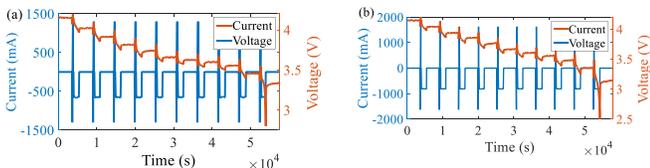


Fig. 7. HPPC load profile (a) LiNMC (b) LiNCA

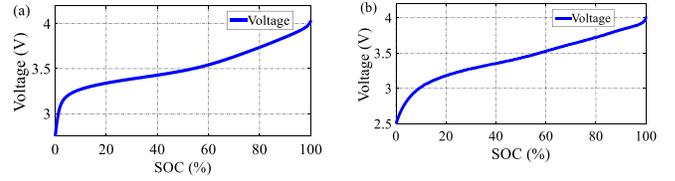


Fig. 8. Relationship between SOC and voltage (a) LiNMC (b) LiNCA

#### E. SOC Effectiveness Measures

The proposed method is validated using numerous performance indicator terms as shown in equations (27-31). The performance of SOC is compared with the actual SOC value. The actual SOC or reference SOC is obtained from ampere hour method.

$$SOC\ error = SOC_a - SOC_{es} \quad (29)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (SOC_{ai} - SOC_{esi})^2 \quad (30)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N (SOC_{ai} - SOC_{esi}) \quad (31)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{SOC_{ai} - SOC_{esi}}{SOC_{ai}} \right| \quad (32)$$

$$SD = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (SOC_{error} - \overline{SOC_{error}})^2} \quad (33)$$

$\overline{SOC_{error}}$  denotes the average value of SOC error.

#### V. EXPERIMENTAL VALIDATION OF SOC ESTIMATION

##### A. Objective Function Assessment and Optimal Parameter

The FA is implemented with 50 population and 500 iterations. The objective function is evaluated by developing the optimization response curve for both static discharge test and HPPC test, as illustrated in Fig. 9 and Fig. 10, respectively.

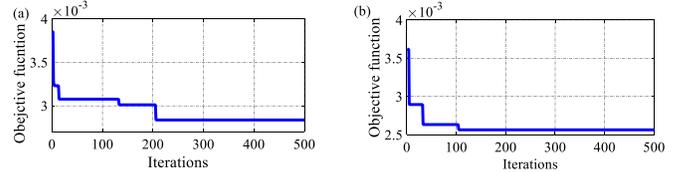


Fig. 9. Optimization response curve for static load profile (a) LiNMC (b) LiNCA

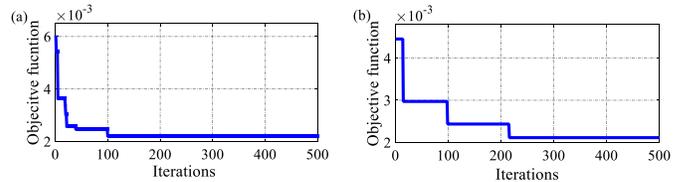


Fig. 10. Optimization response curve for HPPC load profile (a) LiNMC (b) LiNCA

The optimal values of hidden layer and hidden neurons are determined by observing the minimum value of the objective function in the optimization response curve. For instance, in LiNMC and LiNCA battery cell, the minimum value of objective functions of 0.285% and 0.256% are found after 207 and 105 iterations respectively for static discharge test. The corresponding iteration numbers deliver the optimal number of hidden layer and hidden neurons of 4, 6 and 15, 13

respectively. In HPPC load profile, 100, 216 iterations provide the lowest value of the objective function of 0.225%, 0.2114%, delivering the optimal number of hidden layer and hidden neurons 3, 5 and 10, 9 respectively for LiNMC and LiNCA battery cell. The optimal values of hyper-parameters of DRNN attained by FA algorithm is shown in Table II.

TABLE II  
OPTIMAL HYPER-PARAMETERS OF DRNN ALGORITHM

Battery test	Optimal hyper parameters	LiNMC	LiNCA
Static discharge	Hidden layer	4	6
	Hidden neurons	15	13
HPPC	Hidden layer	3	5
	Hidden neurons	10	9

### B. Experimental Results and Comparison Study

The DRNN-FA algorithm based SOC estimation method is validated by the experimental results obtained from static discharge test and HPPC test. The results are evaluated for both LiNMC and LiNCA battery. The performance of DRNN based FA algorithm for SOC estimation is compared with state of art machine learning algorithms including back-propagation neural network (BPNN) and radial basis function neural network (RBFNN). The similar input variables as well as a similar dimension of training and testing dataset are used in both BPNN and RBFNN methods. The network parameters of BPNN and RBFNN are also optimized by FA in order to conduct a fair comparison.

1) *SOC estimation using LiNMC battery:* The DRNN-FA based SOC and SOC error estimation results for static discharge test and HPPC test are presented in Fig. 11 and Fig. 12 respectively. It is noticed from two figures that the SOC line estimated DRNN-FA algorithm is nearly aligned with the reference SOC value which demonstrates the excellent estimation accuracy. The effectiveness of DRNN-FA method is compared with BPNN-FA and RBFNN-FA algorithms using different error rates terms, as shown in Table III. In static discharge cycle, DRNN based FA algorithm achieves RMSE of 1.325% which is a decrease of 29.7% and 13.2% compared to BPNN-FA and RBFNN-FA algorithms, respectively. The results are also improved in MSE, MAE, and MAPE and SD values in DRNN-FA algorithm. For instance, the MAE is computed to be 1.215% that is declined by 19.1% and 4.6% from BPNN-FA and RBFNN-FA algorithms, respectively. Likewise, about 5.3% and 16.5% reductions are noted in DRNN-FA algorithm respectively compared with BPNN-FA and RBFNN-FA algorithm while assessing MAPE. Moreover, in the proposed method, the SOC error is found low and limited to [-2.38%, 4.23%] whereas BPNN-FA and RBFNN-FA algorithms have high SOC error bound of [-5.77%, 8.68%] and [-7.42%, 11.94%] respectively. Similar kind of results is also achieved in the HPPC test where RMSE is dropped by 21.6% and 69.8% compared with BPNN-FA and RBFNN-FA algorithms, respectively. In addition, the DRNN based FA algorithm also obtains relatively small MSE, MAE, MAPE, SD, and SOC error values compared to the other two methods.

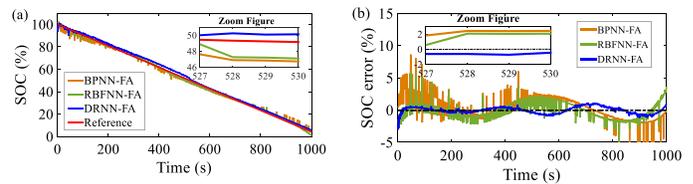


Fig. 11 Experimental results in static discharge test for LiNMC battery (a) SOC (b) SOC error

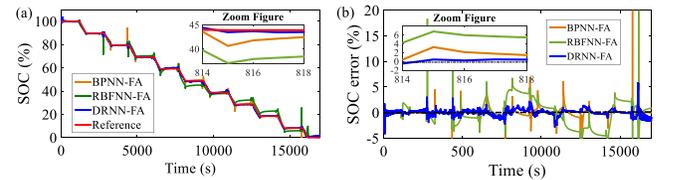


Fig. 12 Experimental results in HPPC test for LiNMC battery (a) SOC (b) SOC error

TABLE III  
PERFORMANCE ASSESSMENT IN LiNMC BATTERY

SOC Method	BPNN-FA		RBFNN-FA		DRNN-FA	
	Static discharge	HPPC	Static discharge	HPPC	Static discharge	HPPC
RMSE (%)	1.884	1.015	1.527	2.634	1.325	0.796
MSE (%)	0.0363	0.011	0.0252	0.0724	0.0164	0.0065
MAE (%)	1.502	0.420	1.274	1.853	1.215	0.512
MAPE (%)	6.157	11.328	6.982	14.746	5.829	8.867
SD	1.868	1.018	1.675	2.642	1.332	0.796
SOC error bound (%)	[-5.77, 8.68]	[-15.17, 21.75]	[-7.42, 11.94]	[-17.89, 27.21]	[-2.38, 4.23]	[-4.72, 5.8]

2) *SOC estimation using LiNCA battery:* Similar battery experiential tests are also conducted with LiNCA battery to check the performance of SOC using DRNN-FA algorithm, as shown in Fig. 13 and Fig. 14. LiNCA battery has stepper SOC-voltage curve compared to LiNMC battery, hence LiNCA achieved fairly lower error rates compared to LiNMC battery. A detailed comparative analysis among the proposed algorithm, BPNN-FA, and RBFNN-FA algorithms is presented in Table IV. The RMSE of the DRNN-FA algorithm in static discharge test is computed to be 1.127% which is a 36.7%, 18.6% reduction from the BPNN-FA and RBFNN-FA algorithms, respectively. The results of DRNN-FA algorithm is also enhanced in terms of MAE and reduced by 31.7% and 13.3% in comparison to BPNN-FA and RBFNN-FA algorithms. Furthermore, the performance of the proposed algorithm is satisfactory while assessing the MSE, MAPE and SD values, SOC error. The results are also found suitable in case of HPPC test obtaining a narrow SOC error bound of [-5.42%, 2.35%], which is lower than the value of [-5.58%, 6.07%] and [-4.79%, 13.65%], attained by BPNN-FA and RBFNN-FA algorithms, respectively.

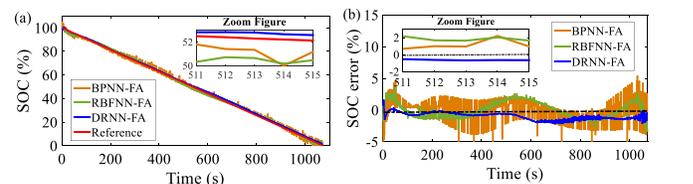


Fig. 13 Experimental results in static discharge test for LiNCA battery (a) SOC (b) SOC error

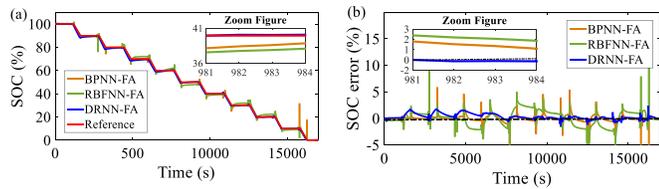


Fig. 14 Experimental results in HPPC test for LiNCA battery (a) SOC (b) SOC error

TABLE IV  
PERFORMANCE ASSESSMENT IN LiNCA BATTERY

SOC Method	BPNN-FA		RBFNN-FA		DRNN-FA	
	Static discharge	HPPC	Static discharge	HPPC	Static discharge	HPPC
RMSE (%)	1.779	0.986	1.384	1.528	1.127	0.596
MSE (%)	0.0347	0.0098	0.0157	0.0258	0.0131	0.0042
MAE (%)	1.428	0.418	1.126	1.237	0.976	0.423
MAPE (%)	5.839	10.623	6.525	11.087	5.788	8.028
SD (%)	1.787	0.992	1.337	1.528	0.597	0.539
SOC error bound (%)	[-5.95, 5.48]	[-9.58, 6.07]	[-3.47, 3.79]	[-4.79, 13.65]	[-2.27, 1.54]	[-5.42, 2.35]

## VI. CONCLUSION

An optimized DRNN algorithm based SOC estimation method is developed for a lithium-ion battery. The computation intelligence of the DRNN algorithm for SOC estimation is enhanced significantly by employing the FA algorithm. The optimum number of hidden layer and hidden neurons are estimated by FA which improves the SOC estimation accuracy and computation speed. The developed algorithm is validated by experiments with two different chemistry of lithium-ion cell, including LiNMC and LiNCA to assess the generalization performance. The proposed method has proved to become an effective SOC algorithm that achieves reasonable accuracy while reducing SOC error under 5% in static discharge test and below 6% in the HPPC test. The comparative validation with BPNN algorithm and RBFNN algorithm also confirms that the DRNN based FA algorithm has low RMSE, MSE, MAE, MAPE and SD during SOC estimation. The experimental results validate the suitability of DRNN-FA algorithm for on-board BMS implementation since few computations is required to estimate SOC in the testing stage. However, there are various issues which will be addressed in future research. The proposed method estimates SOC for a lithium-ion battery, however, the future work will be concerned on the evaluation of DRNN-FA algorithm based SOC estimation for a battery pack of EV. The future research will also assess the performance degradation of lithium-ion batteries with aging effects. Furthermore, the lithium-ion battery SOC performance under different temperatures will be investigated.

## ACKNOWLEDGMENT

This work was supported by the Ministry of Higher Education Grant 20190101LRGS under the Universiti Tenaga Nasional and the Universiti Kebangsaan Malaysia under Grant DIP-2018-020.

## REFERENCES

- [1] M. S. H. Lipu, M. A. Hannan, A. Hussain, M. M. Hoque, P. J. Ker, M. H. M. Saad, and A. Ayob, "A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations," *J. Clean. Prod.*, vol. 205, pp. 115–133, Dec. 2018.
- [2] B. Huang, Z. Pan, X. Su, and L. An, "Recycling of lithium-ion batteries: Recent advances and perspectives," *J. Power Sources*, vol. 399, pp. 274–286, Sep. 2018.

- [3] X. Hu, H. Yuan, C. Zou, Z. Li, and L. Zhang, "Co-Estimation of State of Charge and State of Health for Lithium-Ion Batteries Based on Fractional-Order Calculus," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10319–10329, Nov. 2018.
- [4] L. Zheng, L. Zhang, J. Zhu, G. Wang, and J. Jiang, "Co-estimation of state-of-charge, capacity and resistance for lithium-ion batteries based on a high-fidelity electrochemical model," *Appl. Energy*, vol. 180, pp. 424–434, 2016.
- [5] Y. Zhang, W. Song, S. Lin, and Z. Feng, "A novel model of the initial state of charge estimation for LiFePO 4 batteries," *J. Power Sources*, vol. 248, pp. 1028–1033, 2014.
- [6] P. Shen, M. Ouyang, L. Lu, J. Li, and X. Feng, "The Co-estimation of State of Charge, State of Health, and State of Function for Lithium-Ion Batteries in Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 92–103, Jan. 2018.
- [7] R. Xiong, Y. Zhang, H. He, X. Zhou, and M. G. Pecht, "A Double-Scale, Particle-Filtering, Energy State Prediction Algorithm for Lithium-Ion Batteries," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1526–1538, Feb. 2018.
- [8] B. Xia, Z. Zhang, Z. Lao, W. Wang, W. Sun, Y. Lai, and M. Wang, "Strong Tracking of a H-Infinity Filter in Lithium-Ion Battery State of Charge Estimation," *Energies*, vol. 11, no. 6, p. 1481, Jun. 2018.
- [9] W. He, N. Williard, C. Chen, and 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.*, vol. 62, pp. 783–791, 2014.
- [10] X. Hu, S. E. Li, and Y. Yang, "Advanced Machine Learning Approach for Lithium-Ion Battery State Estimation in Electric Vehicles," *IEEE Trans. Transp. Electrification*, vol. 2, no. 2, pp. 140–149, Jun. 2016.
- [11] J. C. Álvarez Antón, P. J. García Nieto, F. J. de Cos Juez, F. Sánchez Lasheras, M. González Vega, and M. N. Roqueñí Gutiérrez, "Battery state-of-charge estimator using the SVM technique," *Appl. Math. Model.*, vol. 37, no. 9, pp. 6244–6253, 2013.
- [12] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 3, pp. 690–703, Mar. 2017.
- [13] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced Network Anomaly Detection Based on Deep Neural Networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018.
- [14] Y. Chuan-long, Z. Yue-fei, F. Jin-long, and H. Xin-zheng, "A Deep Learning Approach for Intrusion Detection using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 1–1, 2017.
- [15] J. Morton, T. A. Wheeler, and M. J. Kochenderfer, "Analysis of Recurrent Neural Networks for Probabilistic Modeling of Driver Behavior," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1289–1298, May 2017.
- [16] M. A. Hannan, M. S. H. Lipu, A. Hussain, M. H. Saad, and A. Ayob, "Neural Network Approach for Estimating State of Charge of Lithium-Ion Battery Using Backtracking Search Algorithm," *IEEE Access*, vol. 6, pp. 10069–10079, 2018.
- [17] M. S. Hossain Lipu, M. A. Hannan, A. Hussain, and M. H. M. Saad, "Optimal BP neural network algorithm for state of charge estimation of lithium-ion battery using PSO with PCA feature selection," *J. Renew. Sustain. Energy*, vol. 9, no. 6, 2017.
- [18] M. Fairbank, S. Li, X. Fu, E. Alonso, and D. Wunsch, "An adaptive recurrent neural-network controller using a stabilization matrix and predictive inputs to solve a tracking problem under disturbances," *Neural Networks*, vol. 49, pp. 74–86, Jan. 2014.
- [19] X. Fu, S. Li, M. Fairbank, D. C. Wunsch, and E. Alonso, "Training Recurrent Neural Networks With the Levenberg–Marquardt Algorithm for Optimal Control of a Grid-Connected Converter," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 9, pp. 1900–1912, Sep. 2015.
- [20] C. Lv, Y. Xing, J. Zhang, X. Na, Y. Li, T. Liu, D. Cao, and F.-Y. Wang, "Levenberg–Marquardt Backpropagation Training of Multilayer Neural Networks for State Estimation of a Safety-Critical Cyber-Physical System," *IEEE Trans. Ind. Informatics*, vol. 14, no. 8, pp. 3436–3446, Aug. 2018.
- [21] M. K. Kim, "Short-term price forecasting of Nordic power market by combination Levenberg–Marquardt and Cuckoo search algorithms," *IET Gener. Transm. Distrib.*, vol. 9, no. 13, pp. 1553–1563, Oct. 2015.
- [22] R. Toushmalani, "Comparison result of inversion of gravity data of a fault by particle swarm optimization and Levenberg–Marquardt methods," *Springerplus*, vol. 2, no. 1, p. 462, Dec. 2013.
- [23] A. E. Ezugwu and F. Akutsah, "An Improved Firefly Algorithm for the

- Unrelated Parallel Machines Scheduling Problem With Sequence-Dependent Setup Times," *IEEE Access*, vol. 6, pp. 54459–54478, 2018.
- [24] X. Xia, L. Gui, G. He, C. Xie, B. Wei, Y. Xing, R. Wu, and Y. Tang, "A hybrid optimizer based on firefly algorithm and particle swarm optimization algorithm," *J. Comput. Sci.*, vol. 26, pp. 488–500, May 2018.
- [25] H. Su, Y. Cai, and Q. Du, "Firefly-Algorithm-Inspired Framework With Band Selection and Extreme Learning Machine for Hyperspectral Image Classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 1, pp. 309–320, Jan. 2017.
- [26] S. S.-D. Xu, H.-C. Huang, Y.-C. Kung, and S.-K. Lin, "Collision-Free Fuzzy Formation Control of Swarm Robotic Cyber-Physical Systems Using a Robust Orthogonal Firefly Algorithm," *IEEE Access*, pp. 1–1, 2019.
- [27] I. A. Ibrahim and T. Khatib, "A novel hybrid model for hourly global solar radiation prediction using random forests technique and firefly algorithm," *Energy Convers. Manag.*, vol. 138, pp. 413–425, 2017.
- [28] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014.
- [29] R. Zhang, B. Xia, B. Li, L. Cao, Y. Lai, W. Zheng, H. Wang, W. Wang, R. Zhang, B. Xia, B. Li, L. Cao, Y. Lai, W. Zheng, H. Wang, and W. Wang, "State of the Art of Lithium-Ion Battery SOC Estimation for Electrical Vehicles," *Energies*, vol. 11, no. 7, p. 1820, Jul. 2018.
- [30] G. Zubi, R. Dufo-López, M. Carvalho, and G. Pasaoglu, "The lithium-ion battery: State of the art and future perspectives," *Renew. Sustain. Energy Rev.*, vol. 89, pp. 292–308, Jun. 2018.
- [31] Y. Wang, C. Zhang, and Z. Chen, "A method for joint estimation of state-of-charge and available energy of LiFePO4 batteries," *Appl. Energy*, vol. 135, pp. 81–87, 2014.