energies

Article

An Approach to State of Charge Estimation of Lithium-Ion Batteries Based on Recurrent Neural Networks with Gated Recurrent Unit

Chaoran Li, Fei Xiao and Yaxiang Fan *

National Key Laboratory of Science and Technology on Vessel Integrated Power System, Naval University of Engineering, Wuhan 430033, China; lichaoranhg@163.com (C.L.); xfeyninger@gmail.com (F.X.)

* Correspondence: fanyaxiang@126.com; Tel.: +86-1557-599-0344

Received: 26 March 2019; Accepted: 24 April 2019; Published: 26 April 2019

Abstract: State of charge (SOC) represents the amount of electricity stored and is calculated and used by battery management systems (BMSs). However, SOC cannot be observed directly, and SOC estimation is a challenging task due to the battery's nonlinear characteristics when operating in complex conditions. In this paper, based on the new advanced deep learning techniques, a SOC estimation approach for Lithium-ion batteries using a recurrent neural network with gated recurrent unit (GRU-RNN) is introduced where observable variables such as voltage, current, and temperature are directly mapped to SOC estimation. The proposed technique requires no model or knowledge of the battery's internal parameters and is able to estimate SOC at various temperatures by using a single set of self-learned network parameters. The proposed method is evaluated on two public datasets of vehicle drive cycles and another high rate pulse discharge condition dataset with mean absolute errors (MAEs) of 0.86%, 1.75%, and 1.05%. Experiment results show that the proposed method is accurate and robust.

Keywords: state of charge estimation; lithium-ion batteries; gated recurrent unit; recurrent neural networks; neural networks

1. Introduction

Compared with the fossil fuel-based energy, renewable energy has been more and more attractive because it is renewable and cannot be depleted, and it does not contribute to global warming or greenhouse effects. As the energy buffer in renewable energy systems, energy storage could increase the electric power efficiency of the customers and load shifting [1]. Lithium-ion batteries (LiB) have been the first choice of energy storage components [2] due to their advantages of the high energy density, long life cycle, and low self-discharge rate. LiB has a very high energy density, hence a great deal of energy can be stored in it due to the fact that electrodes of LiB are made of lightweight lithium, which is a highly reactive element. High energy density allows a 1 kg LiB to store the same amount of energy as 6 kg batteries made by lead-acid. Additionally, hundreds of charge and discharge cycles can be handled by LiB. Moreover, the charge lost by LiB is as low as 5% per month, whereas the Nickel–Metal Hydride (NIMH) battery has a 20% charge loss per month. Therefore, applications for portable electronics, electrified vehicles, and stationary energy storage are developing and are heavily dependent on LiB technology.

As one of the most important states to be tracked in a battery, state of charge (SOC) is defined as a ratio of the residual capacity to the maximum available capacity, which represents the amount of electricity stored. SOC not only provides reference information for the endurance of the battery but is also the basis of the battery management system (BMS) of the battery pack. BMS is beneficial to the effective utilization of the battery energy and the optimal management of the upper energy system. However, SOC cannot be observed directly and needs to be indirectly estimated by the observable variables such as voltage, current, and temperature. A key issue that must be addressed is how to establish the nonlinear mapping relation between the observable variables and SOC. Additionally, the high dynamic operating conditions of battery will bring another challenge to SOC estimation due to unpredictable charging and discharging.

Traditionally, two widely used SOC estimation methods are the ampere-hour counting method [3] and the open-circuit voltage method [4]. Generally speaking, the former requires the battery to remain resting for a long time to reach balance in order to obtain the open circuit voltage (OCV) versus the SOC curve. On the other hand, for the ampere-hour counting method, the initial SOC needs to be obtained, which has accumulated errors. Both methods have their limitations, and researchers have devoted a lot of effort to put forward new SOC estimation methods, including the model-based method and the data-driven method. With the observable parameters, the model-based methods estimate the SOC state by combining an electrical circuit model (ECM) [5] or an electrochemical model with adaptive filters, such as Kalman filters (KF) [6–9], particle filters [10,11], and adaptive observers [12,13]. Plett et al. [6] first proposed the extended Kalman filter (EKF)-based SOC estimation method for LiB in hybrid electric vehicles. The EKF-based methods [7–9] have been improved with the electrical circuit model or the filter method. In references [10,11] and references [12,13], the particle filter and the adaptive observer were respectively utilized to replace the Kalman filters to improve the accuracy of the SOC. Nevertheless, the SOC estimation accuracy of the model-based method depends very much on the accuracy of the battery model. Actually, researchers have difficulty in characterizing the non-linear characteristics of batteries. In addition, the parameters of different types of batteries are different under varying conditions. There are a lot of parameters to be identified in a battery model, which faces the bottleneck of hardware computation ability. The data-driven methods try to find the non-linear relationship between the observable variables and the battery states from a statistical point of view. These studies have used conventional machine learning techniques such as fuzzy logic [14], support vector machines (SVMs) [15,16], and neural networks [17]. In reference [14], fuzzy logic was used to overcome the problem of battery over-discharge and associated damage resulting from inaccurate estimations of SOC. Alvarez et al. [15] used the application of SVM for SOC estimation of a Li-ion battery. However, the model was validated only in constant pulse charging/discharging working conditions. Similarly, Hu et al. [16] suggested the radial basis function (RBF) kernel to find the optimal parameters for the SVM to apply it to the more complicated work condition. Because the neural network is well suited for modeling complex and nonlinear systems, it has been used for SOC estimation for several years. In reference [17], a feedforward neural network (FNN) was used for SOC estimation in full-electric-vehicle application, and terminal voltage estimation of the batteries was achieved by the SOC estimation results. With the real-time measurement of voltage, current, temperature, and the internal resistance value, Guo et al. [18] proposed another FNN method, called the back propagation neural network (BPNN), to reduce the SOC estimation error significantly. Similarly, Dong et al. [19] used multiple variables as inputs and SOC as the output to train a BPNN. By employing a load-classifying neural network, reference [20] categorized battery operation modes as idle, charge, and discharge, and then estimated the SOC. Recently, in reference [21] and reference [22], Hannan et al. introduced a backtracking search algorithm (BSA) and lighting search algorithm (LSA) to improve the SOC estimation accuracy of the BPNN and nonlinear autoregressive with the exogenous input-based neural network (NARXNN), respectively.

Despite many successful research studies on the FNN-based SOC estimation, significant problems remain unresolved. For instance, the current SOC is not only related to the current observable variables but is also associated with the history observable variables. However, the output of the FNN depends only on the current input, and it is inaccurate to use the FNN process for time sequence problems such as SOC estimation. Moreover, the input of FNN is fixed and can not be arbitrarily changed, which makes it difficult for SOC estimation at any given moment. More recently, the most popular machine learning technique and deep learning [23] (or deep neural network) achieved excellent results for the solution of important problems across a wide spectrum of domains [24–27]. Especially in the field of LiB state estimation, various deep/machine learning approaches have been employed to estimate parameters of LiB, such as the SOC, state of health (SOH), and remaining useful life (RUL) [28,29]. The key to their success is the increase in the amount of available computation and data. As one of the key technologies of deep learning, the recurrent neural network (RNN) [30,31] is quite different from the FNN. It can use the internal state (memory) to learn features and time dependencies from sequential data and has been widely used in natural language processing, time series forecasting, and system modeling. In reference [32], the RNN was used for lead-acid battery modeling and SOC estimation simultaneously. Similarly, Zhao et al. [33] presented a method to combine battery modeling and SOC estimation based on the RNN. The method of Park et al. [34] was also based on the RNN by combining it with the single particle model to build a hybrid battery model.

Different from the RNN-based methods mentioned above, we propose an SOC method based on the gated recurrent unit recurrent neural network (GRU-RNN) to build nonlinear mapping between the observable variables and SOC. Specifically, the GRU-RNN is an improved form of the simple RNN to overcome the short term dependence problem of the simple RNN. The main contributions of our work are as follows: (1) the GRU-RNN can directly characterize the non-linear relationships between voltage, current, temperature, and SOC without the help of a battery model; (2) the GRU-RNN can achieve SOC estimation in various operating conditions with only a set of network parameters, while the other existing methods need different models and parameters for different conditions; (3) the GRU-RNN can self-learn network parameters by adaptive gradient descent algorithms. Compared with electrochemical models and equivalent circuit models that contain differential equations, the GRU-RNN is free from requiring a large amount of work to hand-engineer and parameterize; (4) two public datasets of vehicle drive cycles are used to demonstrate the effectiveness of the proposed method. Moreover, another high rate pulse discharge condition dataset of an 18Ah battery is collected to validate the GRU-RNN's ability for SOC estimation in extreme conditions.

After a brief introduction, Section 2 is a detailed presentation of the proposed approach. In Section 3, three testing datasets recorded respectively under complex discharge conditions, mixed charge-discharge conditions, and high rate pulse discharge conditions are described. Then, the experimental validation of the whole method is exhibited in Section 4. Finally, Section 5 concludes the paper.

2. GRU-RNN for SOC Estimation

The recurrent neural network (RNN) was first put forward in the 1990s in the form of Elman and Jordan networks [35], which are also known as the "simple recurrent networks" (SRN). Different from the FNN, the RNN can use the internal state as the memory of the network. Therefore, the current state is jointly impacted by the current input and the previous state. This structure enables the RNN to deal with the time sequence problem by storing, remembering, and processing past complex signals for a period. RNNs have been widely used in natural language processing, time series forecasting, and system modeling. However, long time series and complex hidden layers may lead to gradients exploding and vanishing during back-propagation processes. This is called the long-term dependencies problem, and plenty of improved RNNs were proposed to solve the problem by designing the gating mechanism to control gradients information propagation. Among all the improved RNNs, the GRU-RNN is not only able to capture long-term sequential dependencies but also has a simple structure. Moreover, compared with other RNNs, it is more robust to vanishing gradients and needs less memory requirements. The illustration of the GRU-RNN cell is shown in Figure 1.

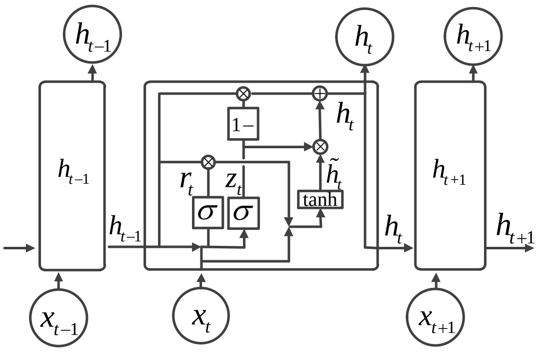


Figure 1. Illustration of the gated recurrent unit memory cell. h_{t-1} is the output of the previous hidden layer node, x_t is input of the current hidden layer node, h_t is the output of the current hidden layer node, z_t and r_t are the output of the update gate and the reset gate, respectively.

The forward propagation of the GRU-RNN is computed by Equations (1) to (4). Among these functions, Equation (1) represents the "update" operation of the GRU-RNN, Equation (2) and Equation (3) represent the "reset" operation of the GRU-RNN, and Equation (4) represents the "output" operation of the GRU-RNN.

$$z_t = \sigma(w_z \cdot [h_{t-1}, x_t] + b_z) \tag{1}$$

$$r_t = \sigma(w_r \cdot [h_{t-1}, x_t] + b_r) \tag{2}$$

$$\widetilde{h_t} = \tanh\Big(w_{\widetilde{h}} \cdot [r_t \odot h_{t-1}, x_t] + b_{\widetilde{h}}\Big) \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t} \tag{4}$$

In Equation (1) to Equation (4), $\odot$ represents an element-wise multiplication; w is the weight parameter; b is the bias parameter; $\sigma(\cdot)$ is the gate activation function, which is set as sigmoid function and shown in Equation (5); $\tanh(\cdot)$ is the output activation function, which is set as tanh function and shown in Equation (6). The derivatives of sigmoid and tanh are the functions of the original function, thus the derivatives can be calculated by the original functions.

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{5}$$

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{6}$$

By introducing the above operations, the GRU-RNN can learn the long-term sequential dependencies. Only the forward propagation is required to obtain the SOC at each timestep during the testing stage. At the end of each forward propagation, the loss function L of the GRU-RNN is calculated as follows:

$$L = \frac{1}{n}\sum_{t=1}^{n}\left|y_t - y_t'\right| \tag{7}$$

where y_t and y_t' are the real value and the estimated value at timestep t, respectively. n is the length of the sequence. Based on the gradient of the loss function, the GRU-RNN updates the weights and the biases by backward propagation. Since the GRU-RNN is a structure through time, it needs a different back propagation method, called the back propagation through time (BPTT), to train the network.

Suppose that the intermediate variable δ_t is defined as:

$$\delta_t = \frac{\partial L}{\partial h_t} \tag{8}$$

The errors in BPTT can be represented as Equation (9), and the errors passed to the upper layer can be represented as Equation (10).

$$\delta_{t-1}^{l} = \delta_{z,t} w_{zh,t} + \delta_{r,t} w_{rh,t} + \delta_{\widetilde{h},t} w_{\widetilde{h}h,t} + \delta_t (1 - z_t) \tag{9}$$

$$\delta_t^{l-1} = \left(\delta_{z,t} w_{zx,t} + \delta_{r,t} w_{rx,t} + \delta_{\widetilde{h},t} w_{\widetilde{h}x,t}\right) \odot f'^{(l-1)} (\chi_t^{l-1}) \tag{10}$$

where $f^{(l-1)}$ is the activation function of layer $l - 1$ neural network nodes, χ_t^{l-1} is the weighted output of layer $l - 1$. $\delta_{z,t}$, $\delta_{r,t}$, $\delta_{\widetilde{h},t}$ can be represented as:

$$\delta_{z,t} = \delta_t \odot \left(\widetilde{h_t} - h_{t-1}\right) \odot z_t \odot (1 - z_t) \tag{11}$$

$$\delta_{r,t} = \delta_t \odot z_t \odot \frac{\partial \widetilde{h_t}}{\partial r_t} \odot r_t \odot (1 - r_t) \tag{12}$$

$$\delta_{\widetilde{h},t} = \delta_t \odot z_t \odot \left(1 - h_t{}^2\right) \tag{13}$$

Gradients of weights and biases can be represented as:

$$\begin{array}{ccc} \frac{\partial L}{\partial w_{zh,t}} = \delta_{z,t} h_{t-1}, & \frac{\partial L}{\partial w_{rh,t}} = \delta_{r,t} h_{t-1}, & \frac{\partial L}{\partial w_{\widetilde{h}h,t}} = \delta_{\widetilde{h},t} h_{t-1} \\ \frac{\partial L}{\partial w_{zx,t}} = \delta_{z,t} x_t, & \frac{\partial L}{\partial w_{rx,t}} = \delta_{r,t} x_t, & \frac{\partial L}{\partial w_{\widetilde{h}x,t}} = \delta_{\widetilde{h},t} x_t \\ \frac{\partial L}{\partial b_{z,t}} = \delta_{z,t}, & \frac{\partial L}{\partial b_{r,t}} = \delta_{r,t}, & \frac{\partial L}{\partial b_{\widetilde{h},t}} = \delta_{\widetilde{h},t} \end{array} \tag{14}$$

The forward propagation, the loss function calculation, and the error back propagation mentioned above constitute a complete training process of the GRU-RNN.

As shown in Figure 2, the inputs of the proposed GRU-RNN based SOC estimation method are the terminal voltage V_t, the current I_t, and the temperature T_t of the battery at timestep t. The output of the GRU-RNN is the SOC$_t$ of the battery at timestep t. $h_t(1), h_t(2), \cdots, h_t(k)$ are hidden layer nodes of the GUR-RNN, where k represents the number of hidden layer nodes and needs to be set in advance. The expansion diagram of a single hidden layer node in the GRU-RNN is shown in Figure 3. Hidden layer nodes take variables $[V_1, I_1, T_1], \ldots, [V_{T-1}, I_{T-1}, T_{T-1}]$, $[V_T, I_T, T_T]$ of time series $t = 1, 2, \cdots, T$ as inputs. The SOC$_t$ at $t = T$ is the output. Other hidden layer nodes are in the same way.

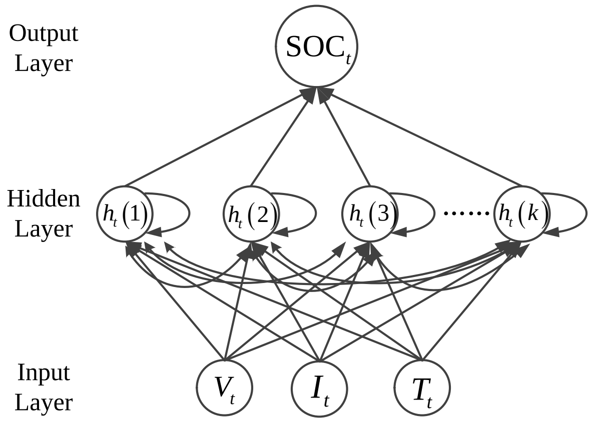


Figure 2. Architecture of the gated recurrent unit recurrent neural network (GRU-RNN) for state of charge (SOC) estimation.

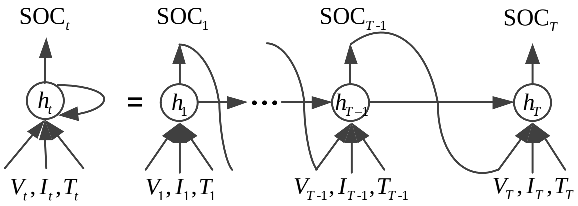


Figure 3. A single hidden layer node in the GRU-RNN unfolded in time.

The processes of the SOC estimation method based on the GRU-RNN are as follows:

Step I: Normalize the testing dataset and divide them into a training set and a validation set.

Step II: Set the parameter of input layer nodes, hidden layer nodes, and output layer nodes of the GRU-RNN; select the form of activation functions, loss functions, and optimization algorithms [36].

Step III: Set the hyperparameters of the GRU-RNN including the timestep, sampling interval, batch size, and iteration. Initialize the weights and biases of the GRU-RNN.

Step IV: Select evaluation function and train the GRU-RNN with the training set. The training set consisting of voltage, current, temperature, and SOC measured values is fed into the initialized GRU-RNN and network parameters can be self-learned according to Equations (1) to (14).

Step V: Validate the effectiveness of the GRU-RNN for SOC estimation with the validation set. The validation set also consists of voltage, current, temperature, and SOC measured values. The voltage, current, and temperature measured values in the validation set are fed into the trained GRU-RNN to obtain SOC estimated values. By inputting the SOC measured values in the validation set and the SOC estimated values of the GRU-RNN into the evaluation function, the effectiveness of the GRU-RNN can be evaluated.

3. Testing Datasets of LiB

In this section, we present the testing data in detail, which are the keys to the GRU-RNN for SOC estimation. In practice, the operating conditions of LiBs are complex and changeable. Therefore, the testing conditions of LiBs should cover the operating conditions as far as possible. In order to satisfy the diversity of operating conditions, three testing datasets of LiBs were selected to verify the validity of the proposed method in this paper. The first two testing datasets are two public datasets, which were collected by testing Panasonic 18650PF battery cells [37] and Samsung 18650-20R battery cells [38] with a series of vehicle drive cycles at three different temperatures. The third testing dataset is a new dataset that was collected under high rate pulsed discharge conditions at three different temperatures.

3.1. The Panasonic 18650PF Dataset

The Panasonic 18650PF dataset [37] was collected by Department of Mechanical Engineering at McMaster University in Ontario, Canada. The Panasonic 18650PF battery cell has nominal voltage and a capacity of 3.6 V and 2.9 Ah, respectively. As shown in Figure 4, nine drive cycles, namely Cycle 1, Cycle 2, Cycle 3, Cycle 4, Supplemental Federal Test Procedure Driving Schedule (US06), the Highway Fuel Economy Test (HWFET), the Urban Dynamometer Driving Schedule (UDDS), the Los Angeles 92 (LA92), and Neural Network (NN) were applied to the same battery cell for 100% battery level at three different temperatures (0 °C, 10 °C, and 25 °C). Among the nine drive cycles, US06, HWFET, UDDS, and LA92 were four common drive cycles of electric vehicles. US06 was a driving condition of electric vehicles with high acceleration. In the HWFET, electric vehicles were tested under 60 miles/hour. The UDDS was used for the driving conditions of light vehicles. The LA92 was developed for emission reduction. Cycles 1, Cycles 2, Cycles 3, Cycles 4, and NN were randomly mixed data composed of US06, HWFET, UDDS, and LA92. Voltage, current, capacity, and cell temperature were recorded in a 0.1-second interval, and more than five million data points were collected. A more detailed description can be found in reference [37], and the maximum values of currents were determined by the LiB operation conditions.

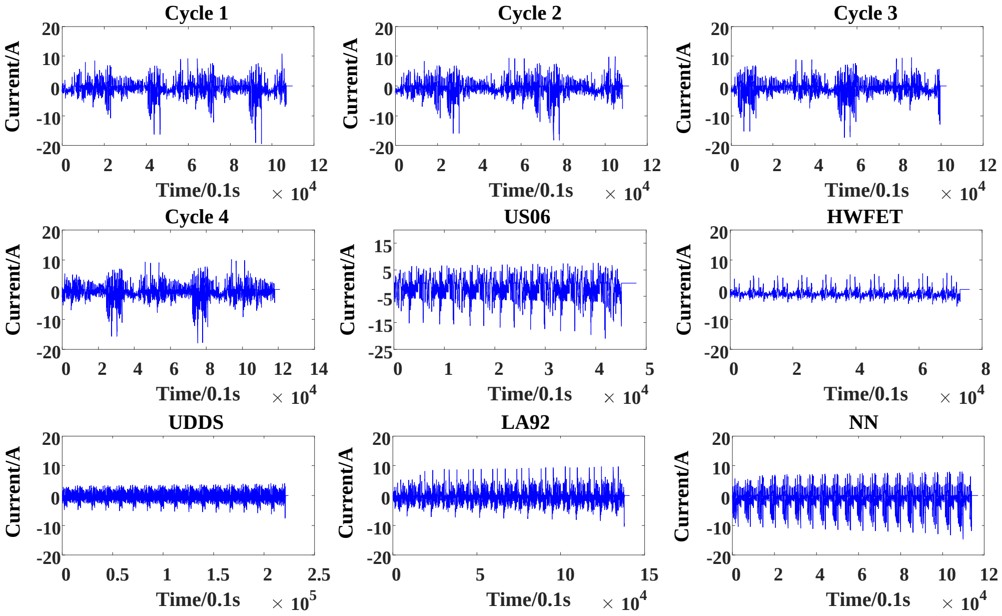


Figure 4. Current profiles measured in Cycle 1, Cycle 2, Cycle 3, Cycle 4, Supplemental Federal Test Procedure Driving Schedule (US06), the Highway Fuel Economy Test (HWFET), the Urban Dynamometer Driving Schedule (UDDS), the Los Angeles 92 (LA92), and Neural Network (NN).

3.2. The Samsung 18650-20R Dataset

The Samsung 18650-20R dataset [38] was collected by the Center for Advanced Life Cycle Engineering (CALCE) at University of Maryland. A 2.9-Ah, 3.2V Samsung 18650-20R cylindrical LiB was tested by four drive cycles—the Beijing Dynamic Stress Test (BJDST), the Federal Urban Driving Schedule (FUDS), US06, and the Dynamic Stress Test (DST). The current profiles of the drive cycles are shown in Figure 5. The maximum values of the currents were determined by the LiB operation conditions. All tests were performed for 80% battery level and 50% battery level at 0 °C, 25 °C, and 45 °C. The voltage, current, capacity, and cell temperature were recorded with a one second time step, and 1.4 million data points were collected. In contrast to the Panasonic 18650PF dataset, the data of charge, pause, and discharge in this dataset were recorded continuously.

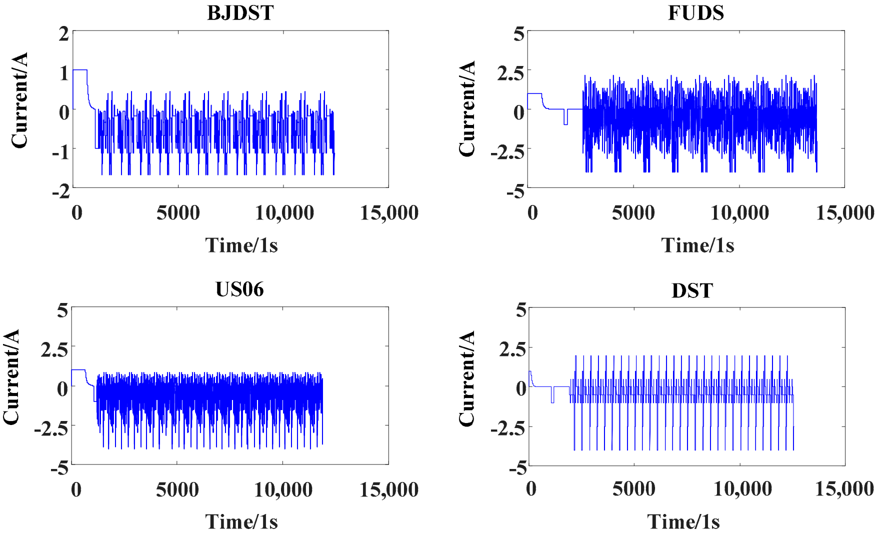


Figure 5. Current profiles measured in the Beijing Dynamic Stress Test (BJDST), the Federal Urban Driving Schedule (FUDS), US06, and the Dynamic Stress Test (DST).

3.3. High Rate Pulse Discharge Condition Dataset Collection

To validate the GRU-RNN's ability for SOC estimation in extreme working conditions, a new dataset of an 18-Ah, 3.65V battery was collected in high rate pulse discharge conditions at the three different temperatures (0 °C, 10 °C, and 25 °C). Similar to the two datasets mentioned above, the test equipment included a host computer, a battery cycler, and a thermal chamber. The test execution steps are: 1) Set the temperature at 0 °C and charge the battery cell fully to 100% SOC using the CC (constant current)-CV (constant voltage) profile at 1 C rate; 2) rest the battery cell for 1 h and discharge the battery cell at a pulse current of 30 C with 0.5 s width every 0.6 s for 80 cycles; 3) rest the battery cell for 1 h and repeat Steps 1 through 2 for temperatures of 45 °C and 25 °C; 4) repeat Steps 1 through 3 two times to obtain three sets of data at each temperature. The voltage, current, capacity, and cell temperature of the discharge stage were recorded in a 0.01-second interval, and 90 thousand data points were collected. This dataset is called the high rate pulse condition dataset throughout the rest of the paper. The corresponding voltage and current profiles of the high rate pulse discharge conditions are shown in Figure 6. From Figure 6, we can observe a significant voltage drop, which is due to the internal resistance of the battery. The differences in temperature lead to differences in the internal resistance of the battery, which further leads to differences in voltage drops [39,40].

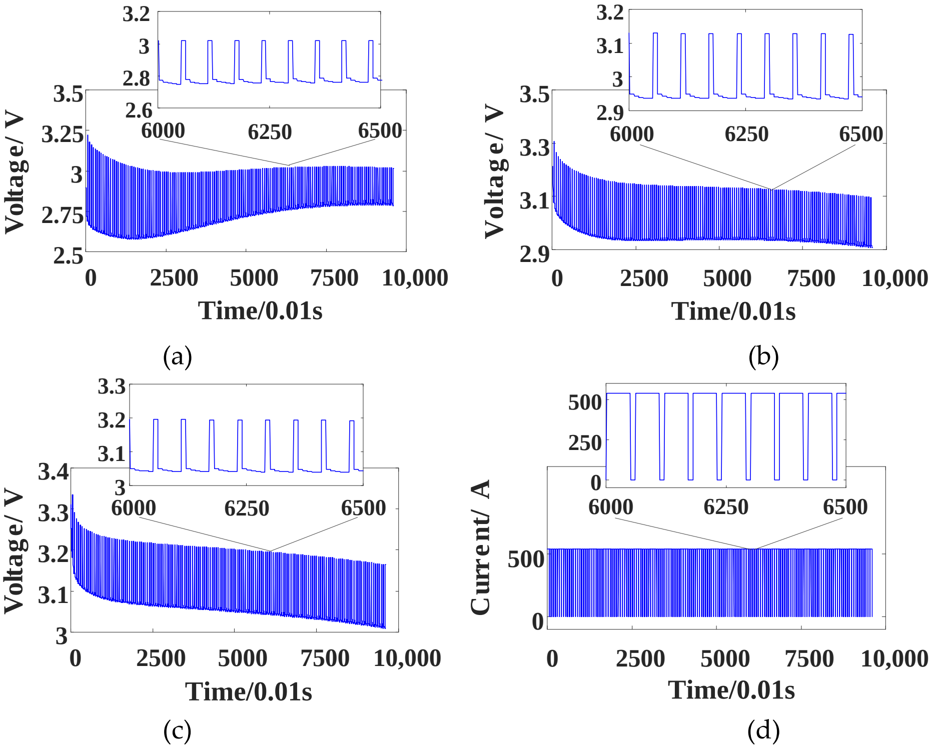


Figure 6. Voltage and current profiles of high rate pulse discharge conditions: (**a**) voltage at 0 °C; (**b**) voltage at 25 °C; (**c**) voltage at 45 °C; (**d**) current.

4. SOC Estimation Results

In this section, we present the experimental settings, evaluation criteria, and experimental results. Specifically, the Panasonic 18650PF dataset, the Samsung 18650-20R dataset, and the high-rate pulse discharge dataset are used to evaluate the performance of the GRU-RNN for SOC estimation in complex and changeable discharge conditions, mixed charge and discharge conditions, and extreme conditions, respectively.

4.1. Experimental Settings

We conduct our experiments on a desktop with Intel Core i7-8700k 3.2 GHz CPU, NVIDIA GeForce GTX 1070Ti (8 GB on-board memory) GPU and 16 GB RAM. The proposed method is implemented in Python and Keras, which uses a Tensorflow backend. Actually, the architecture of the GRU-RNN is very simple, and the GRU-RNN consists of one input layer, one hidden layer, one full connection layer, and one output layer. The nodes of the input layer, the hidden layer, the full connection layer, and the output layer are set as 3, 1000, 50 and 1, respectively. Specifically, the full connection layer transforms the multiple output of the hidden layer into a single SOC value. After setting up the structure of the neural network and the loss function, the network needs to be trained to obtain parameters. The training is a process of minimizing the loss function. Adam is an adaptive optimizer used for minimizing the loss function. The Adam optimizer dynamically adjusts the variations of parameters by using the first-order and second-order matrix estimation of gradients. The parameters of the network are optimized using the Adam optimizer [41] with a learning rate of 0.0001, the first-order momentum attenuation coefficient β_1 of 0.9, the second-order momentum attenuation coefficient β_2 of 0.999, and mini-batches of size 72. In addition, the timestep and the iteration are initialized as 1000 and 100 and then discussed in Section 4.3.3.

Appropriate data normalization can make the training processes of the GRU-RNN more efficient and robust. Moreover, data normalization can remove the negative effect that improves the convergence rate. In this paper, data are normalized to the range [−1,1], as shown in Equation (15)

$$x_{\text{norm}} = \frac{2(x - x_{\min})}{x_{\max} - x_{\min}} - 1 \tag{15}$$

where $x_{\max}$ and $x_{\min}$ are the maximum and the minimum values of data; x represents initial data, and x_{norm} represents the data after normalization.

4.2. Evaluation Criteria

In order to evaluate the accuracy of the GRU-RNN for SOC estimation of battery, the mean absolute error (MAE) and maximum error (MAX) are used for evaluation criteria below.

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|y_t - y'_t\right| \tag{16}$$

$$\text{MAX} = \max\left|y_t - y'_t\right| \tag{17}$$

where y_t and y'_t are the real value and the estimated value at timestep t, respectively. n is the length of the sequence.

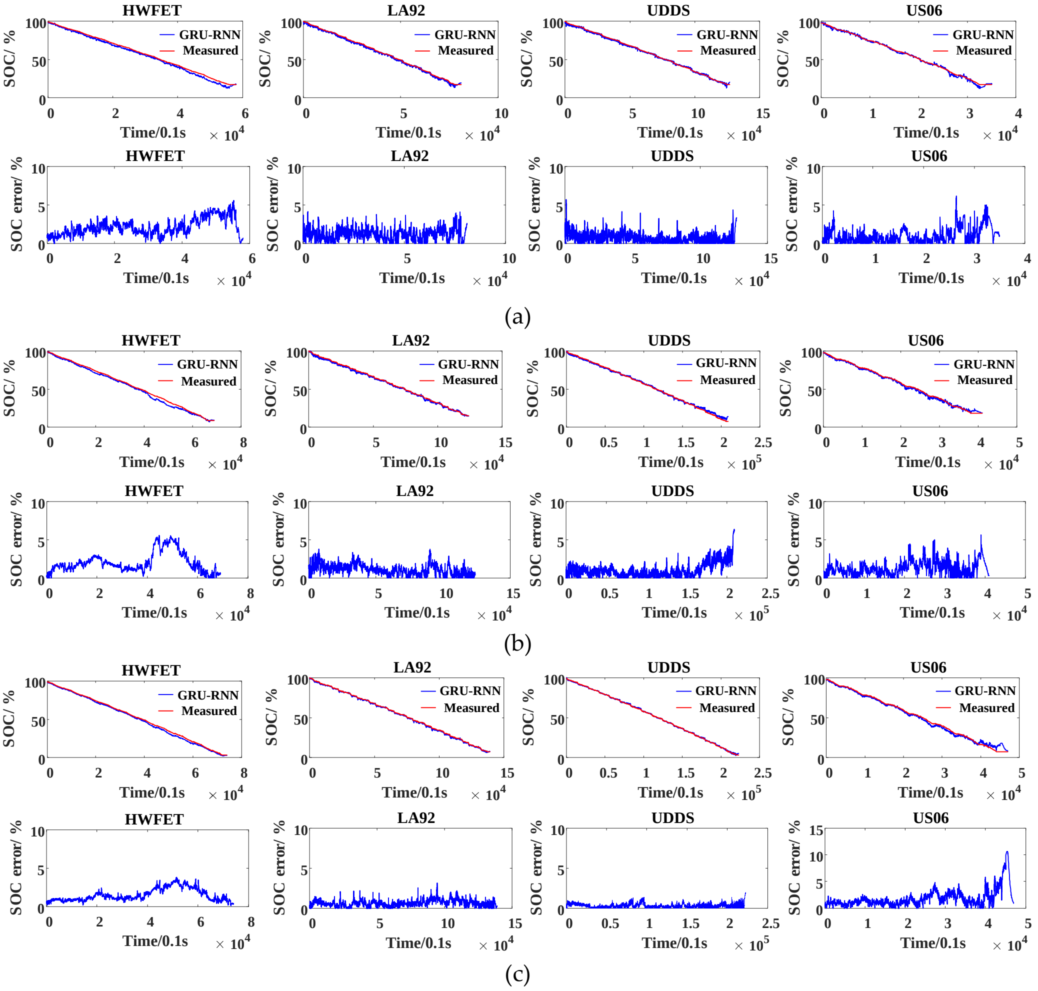
4.3. SOC Estimation under Changeable Discharge Conditions

In this section, the performance of the GRU-RNN for SOC estimation is validated with the Panasonic 18650PF dataset. In these experiments, the effect of single-temperature, multi-temperature, size of training sets, and hyperparameters are discussed. Additionally, some comparisons of SOC estimation between the GRU-RNN and the RNN are made in this section. For the Panasonic 18650PF dataset, the data of Cycle 1, Cycle 2, Cycle 3, Cycle 4, and NN are used for training the GRU-RNN, and the data of US06, HWFET, UDDS, and LA92 are used for validating.

4.3.1. SOC Estimation Trained on Single-Temperature Data

In this experiment, we split the Panasonic 18650PF dataset into three subsets depending on the three different temperatures. The GRU-RNN can self-learn network parameters by the Adam optimizer, which frees researchers from establishments of battery models and identifications of parameters. Then, three single-temperature data are used to evaluate the performance of the GRU-RNN individually.

The SOC estimation curves trained on single-temperature data are shown in Figure 7, and quantitative results are shown in Table 1. In each subgraph of Figure 7, the first row displays the estimated curves, and the second row displays the estimated error curves.



(a)

(b)

(c)

Figure 7. SOC estimation curves trained on single-temperature data: (**a**) 0 °C; (**b**) 10 °C; (**c**) 25 °C.

Table 1. SOC estimation result of the GRU-RNN trained on single-temperature data.

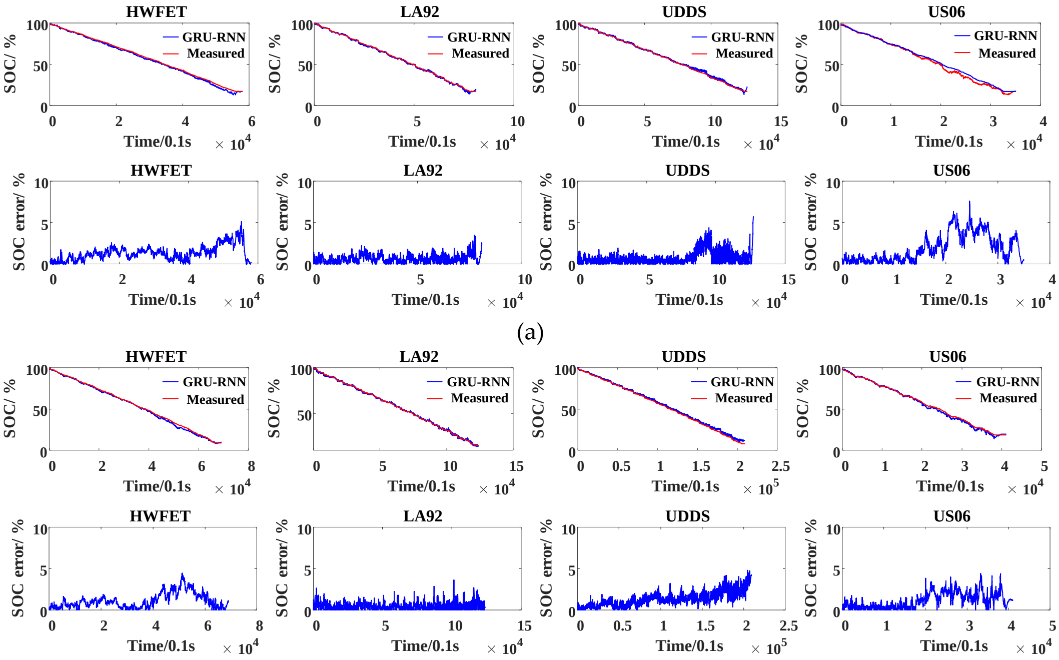
Temperature	Operating Condition	MAE (%)	MAX (%)
0 °C	HWFET	2.12	5.58
	LA92	1.13	4.13
	UDDS	0.71	5.67
	US06	1.01	6.16
10 °C	HWFET	2.02	5.57
	LA92	1.08	3.79
	UDDS	1.05	6.37
	US06	1.27	5.64
25 °C	HWFET	1.50	3.8
	LA92	0.62	3.14
	UDDS	0.32	2.00
	US06	1.86	10.64
Total		1.22	10.64

MAE = mean absolute error, MAX = maximum error.

By simple calculation with the results from Table 1, the average MAE and MAX of SOC estimation results are 1.24%, 1.36%, and 1.08%, and 6.16%, 6.37%, and 10.64% at 0 °C, 10 °C, and 25 °C, respectively. Therefore, the average MAE and MAX of SOC estimation results at three ambient temperatures are 1.22% and 10.64%, which proves that the GRU-RNN can directly characterize the non-linear relationships between voltage, current, temperature, and SOC without the battery model, and the proposed model achieves satisfactory performance in estimating SOC for the single-temperature data.

4.3.2. SOC Estimation Trained on Multi-Temperature Data

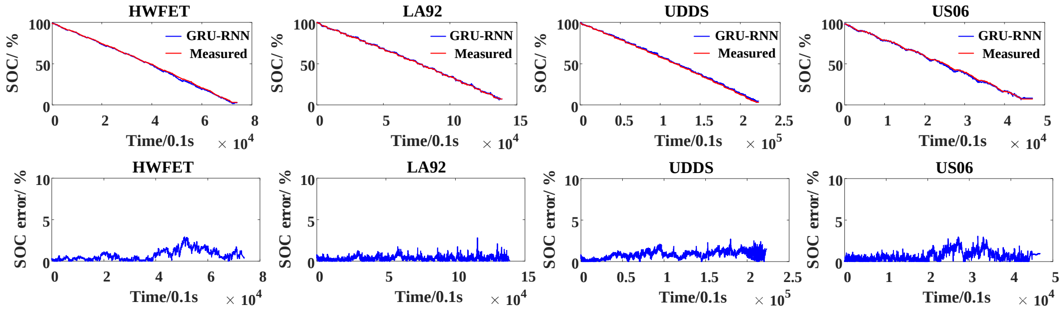
In the last experiment, the performance of the GRU-RNN is evaluated with single-temperature data. However, the network parameters learned with the training set at 0 °C could not be used for SOC estimation with the validation set at 10 °C or 25 °C Therefore, it needs to learn a set of network parameters for the data at each temperature, which seriously increases computational complexity and memory consumption. To address this problem, in this experiment, we utilize the whole training dataset at 0 °C, 10 °C, and 25 °C to train one GRU-RNN and validate this network in HWFET, LA92, UDDS, and US06 at 0 °C, 10°C, and 25 °C The estimation curves are shown in Figure 8. The quantitative results are shown in Table 2. It could be concluded that the average MAE and MAX of SOC estimation results are 0.99%, 0.96%, and 0.63%, and 7.59%, 4.79%, and 3.08% at 0 °C, 10°C, and 25 °C, respectively. The experiment results prove that the GRU-RNN can achieve SOC estimation in various operating conditions with only a set of network parameters. Obviously, the high temperatures result in a low MAE and a low MAX due to reason that the performance of battery increases with the increase in temperature at ambient temperatures. This is because when the ambient temperature approaches room temperature (25 °C), the battery has excellent cycle stability. Compared with the estimation results at single temperature, the average MAE decreases from 1.22% to 0.86%, and the MAX decreases from 10.64% to 7.59% due to the increasing diversity of the training set. For the following experiments, the network parameters are trained with multi-temperature data.



(a)

(b)

Figure 8. *Cont.*



(c)

Figure 8. SOC estimation curves trained on multi-temperature data: (**a**) 0 °C, (**b**) 10 °C, (**c**) 25 °C

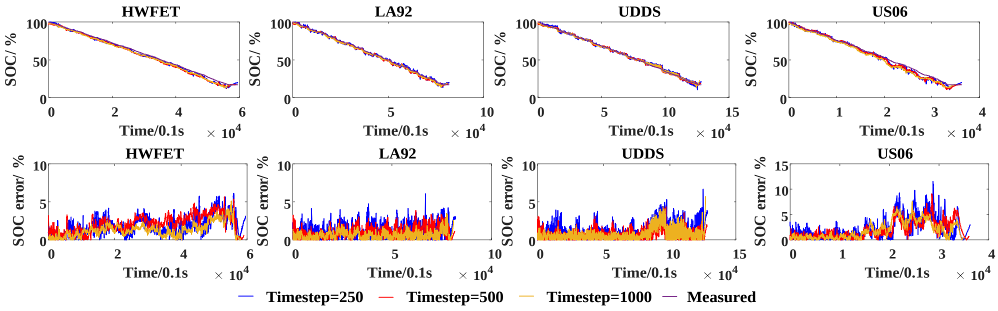
Table 2. SOC estimation result of the GRU-RNN trained on multi-temperature data.

Temperature	Operating Condition	MAE (%)	MAX (%)
0 °C	HWFET	1.31	5.07
	LA92	0.60	3.48
	UDDS	0.77	5.71
	US06	1.27	7.59
10 °C	HWFET	1.11	4.45
	LA92	0.47	3.63
	UDDS	1.12	4.79
	US06	1.13	4.41
25 °C	HWFET	0.71	2.90
	LA92	0.32	2.78
	UDDS	0.82	2.68
	US06	0.68	3.08
Total		0.86	7.59

4.3.3. Influences of Hyperparameters on GRU-RNNs for SOC Estimation

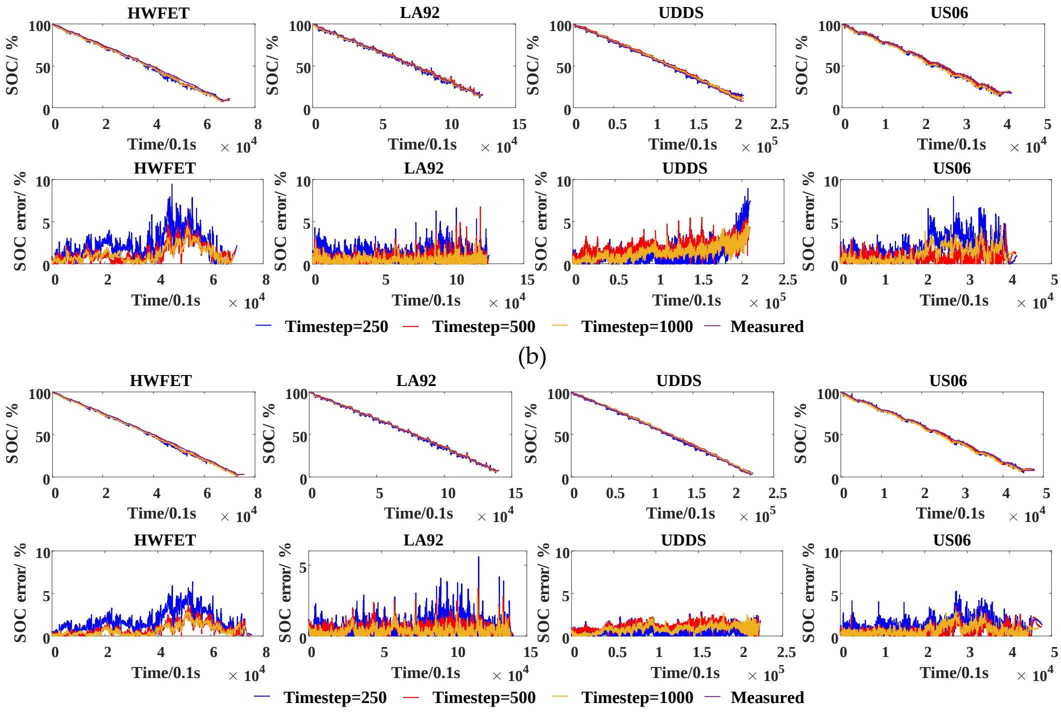
In this sub-section, we analyze the impact of the hyperparameters of the GRU-RNN on the detection results, which specifically include timestep and iteration. In detail, the timestep represents the depth in time of the input layer of the GRU-RNN, and the iteration is the number of the GRU-RNN's training cycles. In the experiments, various values of timestep and iteration are applied, and the SOC estimation results in the forms of MAX and MAE are calculated.

The values of the timestep are set as 250, 500, and 1000, respectively, and other network hyperparameters are consistent with those in Section 4.1. The training times of the GRU-RNN are 12,641 s, 25,012 s, and 51,732 s, respectively. It can be found that the training time is proportional to the value of the timestep. The estimated results are shown in Figure 9 and Table 3. It can be observed that the performance of the GRU-RNN increases with the timestep. Here, the reason is that the input layer of the networks with larger depths in time can learn longer term dependencies of the historical data, which is consistent with the long term dependencies of battery SOC.



(a)

Figure 9. *Cont.*



(b)

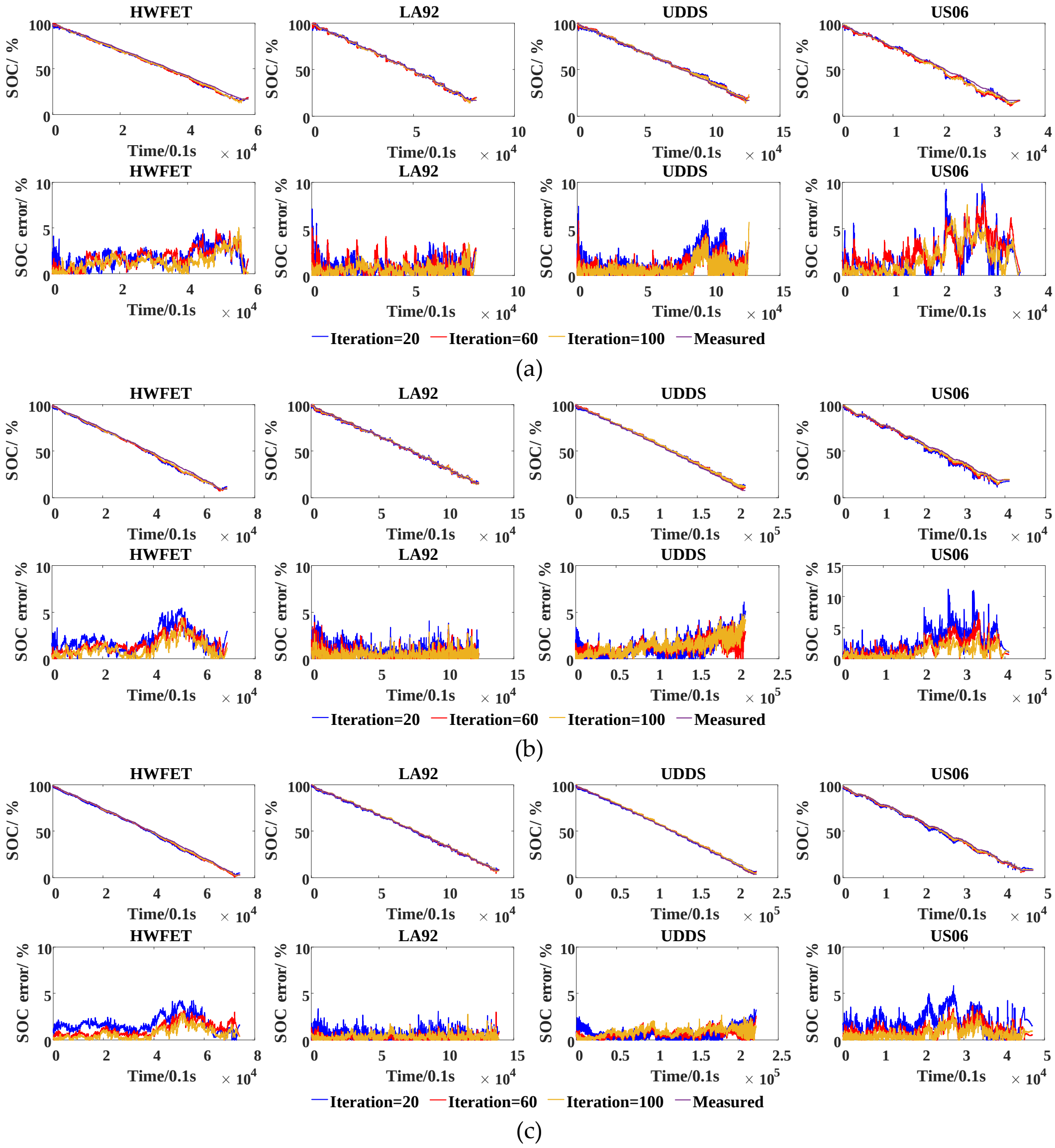
(c)

Figure 9. SOC estimation curves with different values of timestep: (**a**) 0 °C; (**b**) 10 °C; (**c**) 25 °C.

Table 3. SOC estimation results with different values of timestep.

Temperature	Operating Condition	Timestep = 250		Timestep = 500		Timestep = 1000	
		MAE (%)	MAX (%)	MAE (%)	MAX (%)	MAE (%)	MAX (%)
0 °C	HWFET	1.83	6.11	2.12	5.63	1.31	5.07
	LA92	1.09	6.04	1.02	3.87	0.60	3.48
	UDDS	1.01	6.67	0.72	4.31	0.77	5.71
	US06	2.22	11.53	1.96	9.10	1.27	7.59
10 °C	HWFET	2.16	9.4	1.22	5.25	1.11	4.45
	LA92	1.12	6.61	0.61	6.74	0.47	3.63
	UDDS	1.19	8.91	1.79	5.49	1.12	4.79
	US06	1.65	7.95	0.78	4.68	1.13	4.41
25 °C	HWFET	1.76	6.38	0.80	3.64	0.71	2.90
	LA92	0.66	5.58	0.51	3.41	0.32	2.78
	UDDS	0.52	2.86	1.11	2.83	0.82	2.68
	US06	1.18	5.29	0.62	3.57	0.68	3.08
Total		1.37	11.53	1.11	9.10	0.86	7.59

Another experiment is performed to measure the impact on the estimation results of different values of iteration. In detail, the values of iteration are set as 20, 40, 60, 80, and 100, respectively, and other network hyperparameters are consistent with those in Section 4.1. The training times of the GRU-RNN are 9979 s, 20,575 s, 29,623 s, 40,982 s and 51,732 s, respectively. Additionally, it can be found that the training time is proportional to the value of iteration. The estimated curves and the corresponding bar graph of the estimation results are shown in Figures 10 and 11. It is immediately observable that the more iteration is exposed, the better the accuracy is. The MAE decreases from 1.46% to 0.86% and the MAX decreases from 11.23% to 7.59% when the values of iteration increase from 20 to 100. This is mainly because the value of iteration increases, thus the training data sets are randomly reused by the GRU-RNN, which increases the diversity of training samples.



(a)

(b)

(c)

Figure 10. SOC estimation curves with different values of iteration: (**a**) 0 °C; (**b**) 10 °C; (**c**) 25 °C.

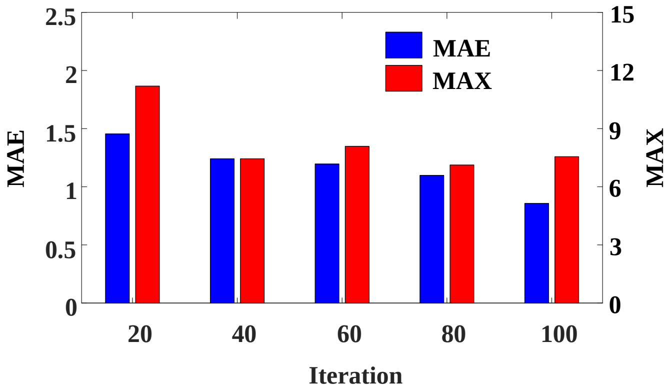
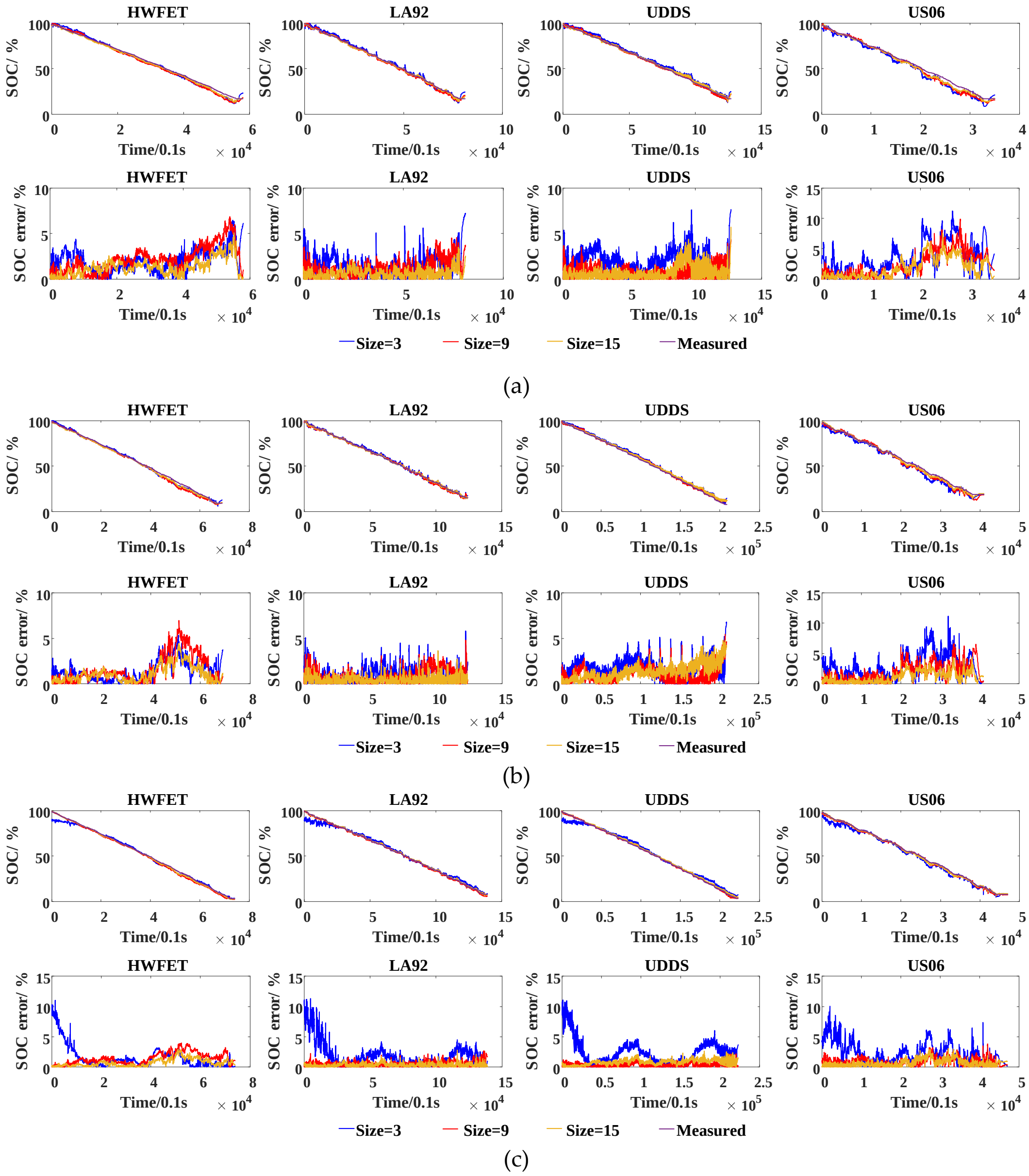


Figure 11. Performances at different values of iteration.

4.3.4. Influences of the Size of Training Data on GRU-RNN for SOC Estimation

In order to analyze the influences of the training data size on the GRU-RNN for SOC estimation, we record the estimation result of the GRU-RNN trained on different sizes of mixed drive cycles. In detail, the GRU-RNN is trained on one to five mixed drive cycles at various temperatures. Accordingly, the sizes of the training data are three, six, nine, 12, and 15, respectively. The other network hyperparameters are consistent with those in Section 4.1. For display purposes, only the estimated curves with sizes of three, nine, and 15 are shown in Figure 12.



(a)

(b)

(c)

Figure 12. SOC estimation curves with different sizes of training data: (**a**) 0 °C; (**b**) 10 °C; (**c**) 25 °C.

The corresponding bar graphs of the estimation results are shown in Figure 13. It is immediately observable that, in this case, the performance of the GRU-RNN increases with the sizes of training data. It should also be noted that the GRU-RNN can achieve a MAE below 1.5% when training is conducted on nine or more training data. This means the proposed GRU-RNN is robust for SOC estimation when the training data is inadequate.

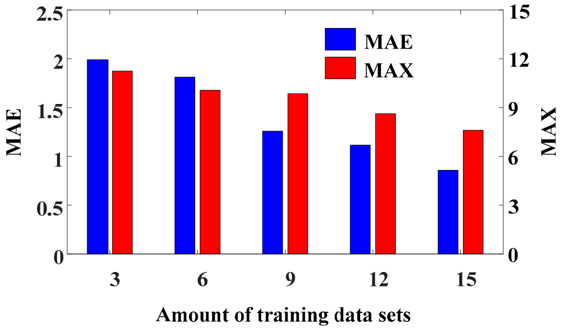
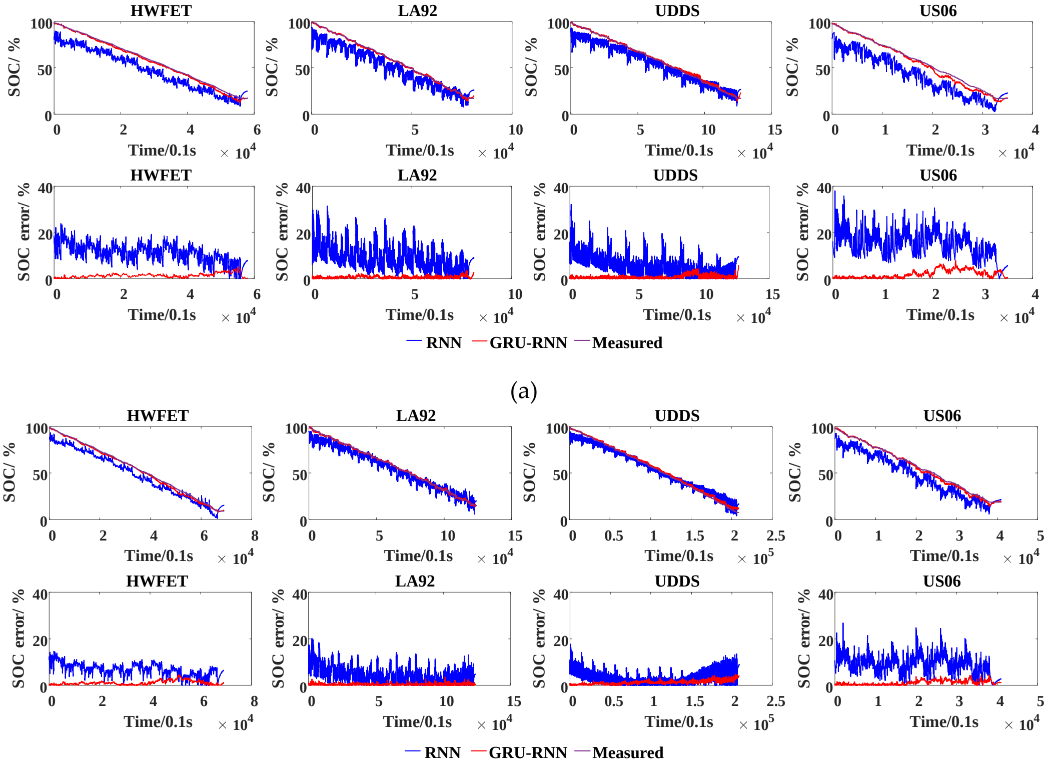


Figure 13. Performances at different sizes of training data.

4.3.5. Comparisons between GRU-RNN and RNN

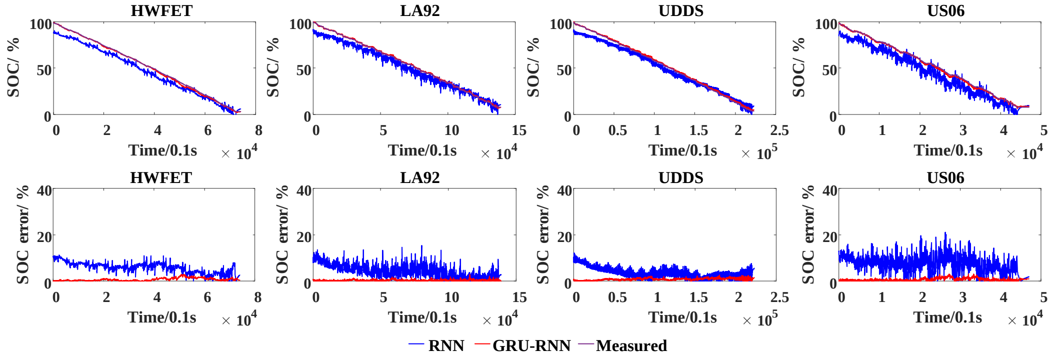
In order to demonstrate the advantages of the adopted GRU-RNN, we replace the GRU-RNN with an original RNN. The same network architecture as the GRU-RNN is adopted by the RNN for a fair comparison. The training times of the RNN and the GRU-RNN are 30,152 s and 51,732 s, respectively. The GRU-RNN takes a longer time to train the network parameters than the RNN due to a more complex structure, as shown in Figure 1. Figure 14 compares the performance of the RNN and the GRU-RNN. It is clear that, compared with the RNN (blue curve), the GRU-RNN (red curve) has smaller SOC errors. The detailed comparison results are shown in Table 4. It can be seen that the GRU-RNN outperforms the RNN in both the MAE and the MAX. The GRU-RNN is more suitable than the RNN for SOC estimation with long-term dependencies. The SOC of LiBs is a sequence that increases over time, and the historical sequences have influence on the current sequences. In order to accurately estimate the current SOC of LiBs, as many historical sequences as possible should be taken into account. The RNN cannot deal with long sequences well because of exploding gradient and vanishing gradient. However, the GRU-RNN can realize the control of the current input and the historical state by GRU, as introduced in Section 2.



(a)

(b)

Figure 14. *Cont.*



(c)

Figure 14. Performance comparison of the recurrent neural network (RNN) and the GRU-RNN: (**a**) 0 °C; (**b**) 10 °C; (**c**) 25 °C.

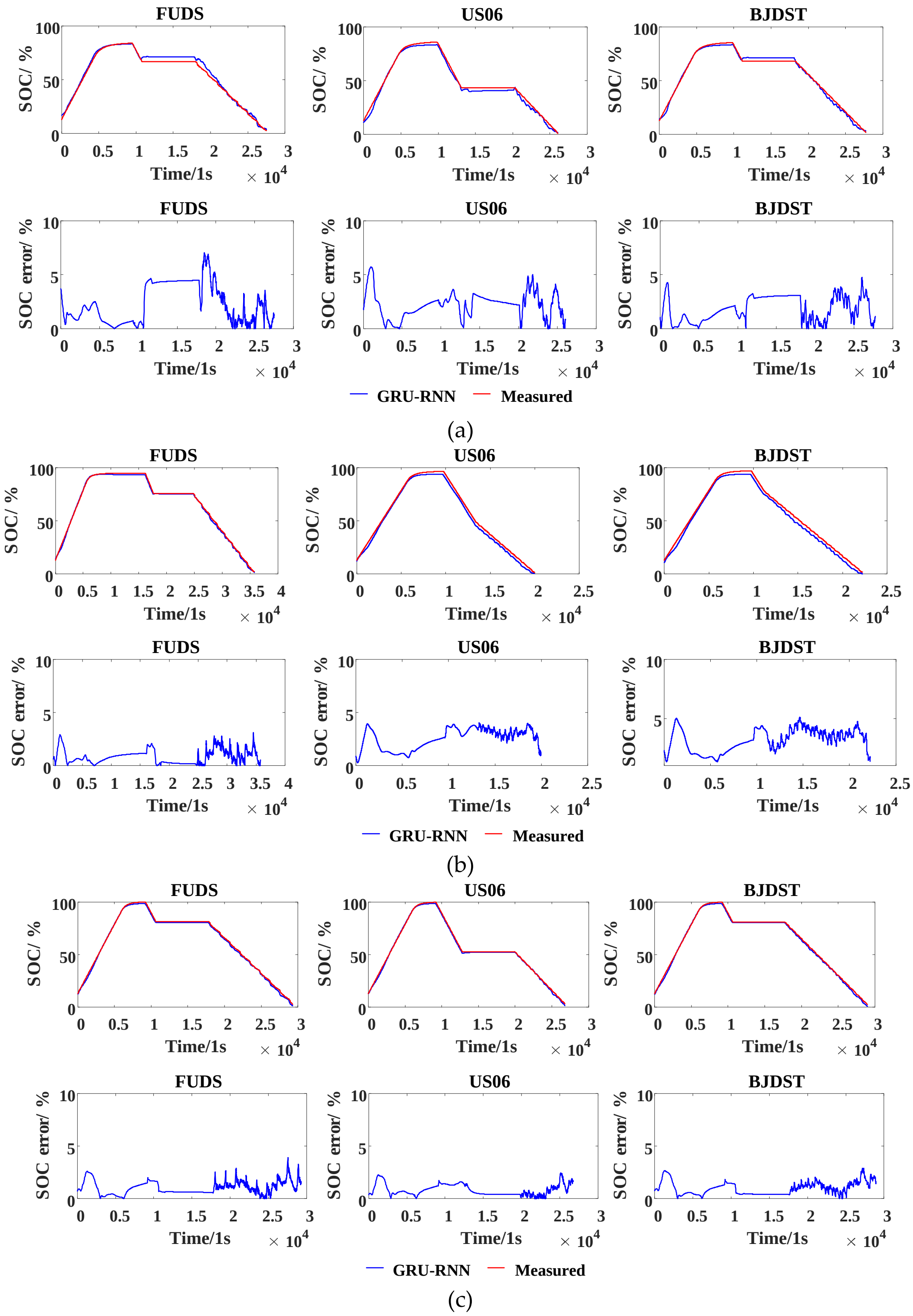
Table 4. Quantitative comparison of the RNN and the GRU-RNN.

Temperature	Operating Condition	RNN		GRU-RNN	
		MAE (%)	MAX (%)	MAE (%)	MAX (%)
0 °C	HWFET	11.06	23.7	1.31	5.07
	LA92	7.84	31.27	0.60	3.48
	UDDS	4.87	32.06	0.77	5.71
	US06	14.75	37.92	1.27	7.59
10 °C	HWFET	6.75	14.58	1.11	4.45
	LA92	4.31	20.11	0.47	3.63
	UDDS	3.32	17.84	1.12	4.79
	US06	8.96	26.89	1.13	4.41
25 °C	HWFET	5.69	11.29	0.71	2.90
	LA92	4.08	15.29	0.32	2.78
	UDDS	3.48	12.15	0.82	2.68
	US06	7.32	21.08	0.68	3.08
Total		6.87	37.92	0.86	7.59

4.4. SOC Estimation under Mixed Charge-Discharge Conditions

In order to validate the performance of the proposed GRU-RNN for SOC estimation in whole working states of battery, we evaluate the performance of the proposed method on the Samsung 18650-20R dataset, in which the data of charge, pause, and discharge are recorded continuously. A detailed description of the Samsung 18650-20R dataset can be found in Section 3.2. Two sets of data at each ambient temperature under FUDS, US06, and BJDST are used in this experiment. Half of the data are used for training, and the other half are used for validating.

The SOC estimation curves are calculated and compared with the measured SOC estimation curves in Figure 15. The curves are similar but slightly differ from each other. The quantitative results are shown in Table 5 with a total average MAE of 1.75% and a maximum MAX of 7.04%. This proves that the GRU-RNN can accurately estimate the SOC under mixed charge-discharge conditions.



(a)

(b)

(c)

Figure 15. SOC estimation curves under mixed charge-discharge conditions: (**a**) 0 °C; (**b**) 25 °C; (**c**) 45 °C.

Table 5. SOC estimation results under mixed charge-discharge conditions.

Temperature	Operating Condition	MAE (%)	MAX (%)
0 °C	FUDS	2.28	7.04
	US06	2.19	5.72
	BJDST	1.84	4.73
25 °C	FUDS	0.86	3.13
	US06	2.56	4.03
	BJDST	3.17	5.11
45 °C	FUDS	1.05	3.91
	US06	0.83	2.43
	BJDST	0.98	2.87
Total		1.75	7.04

4.5. SOC Estimation under High Rate Pulse Discharge Conditions

The last experiment is conducted under high rate pulse discharge conditions with the high rate pulse discharge condition dataset to validate the GRU-RNN's ability for SOC estimation under extreme conditions. A set of data for each temperature is used for training, and another set of data for each temperature is used for validating. The SOC estimation curves and the corresponding quantitative results are shown in Figure 16 and Table 6, respectively. It is clear that the GRU-RNN achieves excellent results with MAE = 1.05% and MAX = 2.22% on the high rate pulse discharge condition dataset, which indicates the feasibility and validity of this method.

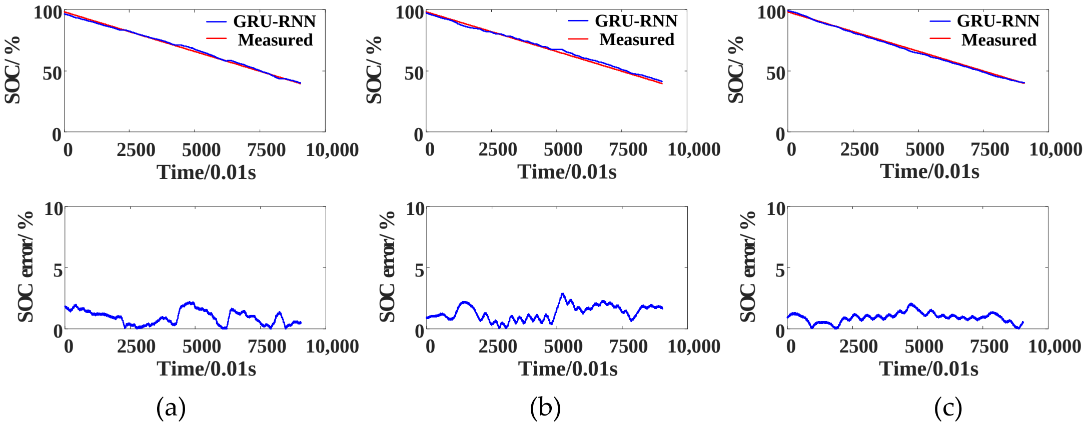


Figure 16. SOC estimation curves under high rate pulse discharge conditions: (**a**) 0 °C; (**b**) 25 °C; (**c**) 45 °C.

Table 6. SOC estimation result under high rate pulse discharge conditions.

Temperature	MAE (%)	MAX (%)
0 °C	0.93	1.97
25 °C	1.32	2.22
45 °C	0.91	2.04
Total	1.05	2.22

In summary, the proposed SOC estimation approach for LiBs using the GRU-RNN has been evaluated in complex and changeable discharge conditions, mixed charge and discharge conditions, and extreme conditions by the Panasonic 18650PF dataset, the Samsung 18650-20R dataset, and the high-rate pulse discharge dataset. The MAEs of the experiment results are 0.86%, 1.75%, and 1.05%, and the MAXs are 7.59%, 7.04%, and 2.22%. This proves that the proposed method has good abilities in accuracy and robustness. In addition, the GRU-RNN can self-learn network parameters by the

Adam optimizer, which frees researchers from the establishments of battery models and identifications of parameters.

5. Conclusions

In this work, an accurate and robust SOC estimation approach is developed for LiBs. The proposed strategy is based on a machine learning framework and leverages the GRU-RNN to establish the nonlinear mapping relation between the observable variables and SOC. Both the qualitative and the quantitative results on three challenging datasets prove the feasibility and advantages of the proposed method. Two public datasets of vehicle drive cycles and another high rate pulse discharge condition dataset are used to evaluate the performance of the GRU-RNN for SOC estimation in complex and changeable discharge conditions, mixed charge and discharge conditions, and extreme conditions, respectively. In addition, the following works are accomplished in this paper: (a) the proposed GRU-RNN method can directly characterize the non-linear relationships between voltage, current, temperature, and SOC without battery models; (b) the proposed GRU-RNN method is proven to have the ability to estimate SOC at various temperatures by using a single set of network parameters; (c) the proposed GRU-RNN method is proven to self-learn network parameters without requiring a great deal of work to hand-engineer and parameterize; (d) the influences of the GRU-RNN hyperparameters and training data size on the performance for SOC estimation are analyzed. The experimental results demonstrate the performance of the GRU-RNN increases with the hyperparameters such as timestep and iteration as well as the training data size. The increase of timestep represents the increase of sequence span during the GRU-RNN training, and the increase of iteration represents the increase of diversity of the GRU-RNN training samples. It is also proven that the model still has high estimation accuracy, even though the amount of data is small; (e) the comparison results between the RNN and the GRU-RNN show that the GRU-RNN can achieve a higher accuracy and overcome the problem of long-term dependencies in the RNN. In summary, the proposed SOC estimation method for LiBs based on the GRU-RNN has been widely validated and achieved good results. To extend this work, the future plan might entail conducting battery lifetime experiments to obtain battery aging data for evaluating the accuracy of the proposed method after a large number of cycles and optimizing the datasets for testing with complete mission profiles. In addition, future work will design the proposed method in the prototype hardware implementation for the BMS.

Author Contributions: Conceptualization, C.L., F.X. and Y.F.; Data curation, C.L.; Formal analysis, F.X.; Funding acquisition, F.X.; Investigation, Y.F.; Methodology, C.L. and Y.F.; Project administration, F.X.; Resources, F.X.; Software, C.L. and Y.F.; Supervision, F.X.; Validation, C.L. and Y.F.; Visualization, C.L.; Writing—original draft, C.L.; Writing—review & editing, F.X. and Y.F.

Funding: This research was funded by National Defense Science Innovation Zone Project and National Natural Science Foundation, grant number 51807200.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Wade, N.S.; Taylor, P.C.; Lang, P.D.; Jones, P.R. Evaluating the benefits of an electrical energy storage system in a future smart grid. *Energy Policy* **2010**, *38*, 7180–7188. [CrossRef]
2. Dunn, B.; Kamath, H.; Tarascon, J.M. Electrical energy storage for the grid: a battery of choices. *Science* **2011**, *334*, 928–935. [CrossRef] [PubMed]
3. Liu, Z.; Li, Z.; Zhang, J.; Su, L.; Ge, H. Accurate and Efficient Estimation of Lithium-Ion Battery State of Charge with Alternate Adaptive Extended Kalman Filter and Ampere-Hour Counting Methods. *Energies* **2019**, *12*, 757. [CrossRef]
4. Hannan, M.A.; Lipu, M.S.H.; 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*, 834–854. [CrossRef]

5. Lai, X.; Zheng, Y.; Sun, T. A comparative study of different equivalent circuit models for estimating state-of-charge of lithium-ion batteries. *Electrochim. Acta* **2018**, *259*, 566–577. [CrossRef]
6. Plett, G.L. Extended kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Background. *J. Power Sources* **2004**, *134*, 252–261. [CrossRef]
7. Hua, 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*, 209–219. [CrossRef]
8. He, Z.W.; Gao, M.Y.; Wang, C.S.; Wang, L.Y.; Liu, Y.Y. Adaptive state of charge estimation for Li-ion batteries based on an unscented kalman filter with an enhanced battery model. *Energies* **2013**, *6*, 4134–4151. [CrossRef]
9. Li, J.; Barillas, J.K.; Guenther, C.; Danzer, M.A. A comparative study of state of charge estimation algorithms for LiFePO4 batteries used in electric vehicles. *J. Power Sources* **2013**, *230*, 244–250. [CrossRef]
10. He, H.; Xiong, R.; Guo, H.; Li, S. Comparison study on the battery models used for the energy management of batteries in electric vehicles. *Energy Convers. Manag.* **2012**, *64*, 113–121. [CrossRef]
11. Rubio, J.; Pascual-Iserte, A. Energy-aware broadcast multiuse-MIMO precoder design with imperfect channel and battery knowledge. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 3137–3152. [CrossRef]
12. Zheng, L.; Jiang, J.; Wang, Z.; Zhao, T. Embedded implementation of SOC estimation based on the Luenberger observer technique. In Proceedings of the IEEE Conference and Expo Transportation Electrification Asia-Pacific, Beijing, China, 31 August–3 September 2014.
13. Ning, B.; Xu, J.; Cao, B.; Xu, G. A sliding mode observer SOC estimation method based on parameter adaptive battery model. *Energy Procedia* **2016**, *88*, 619–626. [CrossRef]
14. Li, S.G.; Sharkh, S.M.; Walsh, F.C.; Zhang, C.N. Energy and battery management of a plug-in series hybrid electric vehicle using fuzzy logic. *IEEE Trans. Veh. Technol.* **2011**, *60*, 3571–3585. [CrossRef]
15. Alvarez, A.J.C.; Garcia, N.P.J.; Blanco, V.C.; Vilan, J.A. Support vector machines used to estimate the battery state of charge. *IEEE Trans. Power Electr.* **2013**, *28*, 5919–5926. [CrossRef]
16. Hu, J.N.; Hu, J.J.; Lin, H.B.; Li, X.P.; Jiang, C.L.; Qiu, X.H.; Li, W.S. State-of-charge estimation for battery management system using optimized support vector machine for regression. *J. Power Sources* **2014**, *269*, 682–693. [CrossRef]
17. Eddahech, A.; Briat, O.; Vinassa, J.M. Adaptive voltage estimation for EV Li-ion cell based on artificial neural networks state-of-charge meter. In Proceedings of the IEEE International Symposium on Industrial Electronics, Hangzhou, China, 28–31 May 2012.
18. Guo, Y.; Zhao, Z.; Huang, L. SoC estimation of Lithium battery based on improved BP neural network. *Energy Procedia* **2017**, *105*, 4153–4158. [CrossRef]
19. Dong, C.; Wang, G. Estimation of power battery SOC based on improved BP neural network. In Proceedings of the IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014.
20. Tong, S.; Lacap, J.H.; Park, J.W. Battery state of charge estimation using a load-classifying neural network. *J. Energy Storage* **2016**, *7*, 236–243. [CrossRef]
21. Hannan, M.A.; Lipu, M.S.H.; Hussain, A.; Saad, M.H.M. Neural network approach for estimating state of charge of Lithium-ion battery using backtracking search algorithm. *IEEE Access* **2018**, *6*, 10069–10079. [CrossRef]
22. Lipu, M.S.H.; Hannan, M.A.; Hussain, A.; Saad, M.H.M. State of charge estimation for Lithium-ion battery using recurrent NARX neural network model based lighting search algorithm. *IEEE Access* **2018**, *6*, 28150–28161. [CrossRef]
23. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
24. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Driessche, G.V.D.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef]
25. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [CrossRef]
26. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
27. Banino, A.; Barry, C.; Uria, B.; Blundell, C.; Lillicrap, T.; Mirowski, P.; Pritzel, A.; Chadwick, M.J.; Degris, T.; Modayil, J.; et al. Vector-based navigation using grid-like representations in artificial agents. *Nature* **2018**, *557*, 429–433. [CrossRef]

28. Zhang, Y.; Xiong, R.; He, H.; Pecht, M.G. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Trans. Veh. Technol.* **2018**, *67*, 5695–5705. [CrossRef]
29. Khumprom, P.; Yodo, N. A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm. *Energies* **2019**, *12*, 660. [CrossRef]
30. Williams, R.J.; Zipser, D. *A learning algorithm for continually running fully recurrent neural networks*; MIT Press: Cambridge, MA, USA, 1989.
31. Sutskever, I.; Martens, J.; Hinton, G.E. Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011.
32. Capizzi, G.; Bonanno, F.; Tina, G.M. Recurrent neural network-based modeling and simulation of lead-acid batteries charge–discharge. *IEEE Trans. Energy Convers.* **2011**, *26*, 435–443. [CrossRef]
33. Zhao, R.; Kollmeyer, P.J.; Lorenz, R.D.; Jahns, T.M. A compact unified methodology via a recurrent neural network for accurate modeling of lithium-ion battery voltage and state-of-charge. In Proceedings of the IEEE Energy Conversion Congress and Exposition (ECCE), Cincinnati, OH, USA, 1–5 October 2017.
34. Park, S.; Zhang, D.; Moura, S. Hybrid electrochemical modeling with recurrent neural networks for li-ion batteries. In Proceedings of the American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017.
35. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
36. Ian, G. *Deep learning*; Posts & Telecom Press: Beijing, China, 2017.
37. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Emadi, A. State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach. *J. Power Sources* **2018**, *400*, 242–255. [CrossRef]
38. 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. [CrossRef]
39. Hande, A. Internal battery temperature estimation using series battery resistance measurements during cold temperatures. *J. Power Sources* **2006**, *158*, 1039–1046. [CrossRef]
40. Barai, A.; Widanage, W.D.; McGordon, A.; Jennings, P. The influence of temperature and charge-discharge rate on open circuit voltage hysteresis of an LFP Li-ion battery. In Proceedings of the IEEE Transportation Electrification Conference and Expo (ITEC), Dearborn, MI, USA, 27–29 June 2016.
41. Jiang, S.; Guo, K.; Liao, J.; Zheng, G. Solving Fourier ptychographic imaging problems via neural network modeling and TensorFlow. *Biomed. Opt. Express* **2018**, *9*, 3306–3319. [CrossRef]