IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 23, 2019, accepted July 9, 2019, date of publication July 10, 2019, date of current version July 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2928037

Convolutional Gated Recurrent Unit–Recurrent Neural Network for State-of-Charge Estimation of Lithium-Ion Batteries

ZHELIN HUANG[1], FANGFANG YANG[1], FAN XU[1], XIANGBAO SONG[2], AND KWOK-LEUNG TSUI[1]
[1]School of Data Science, City University of Hong Kong, Hong Kong, China
[2]Googol Technology (Shenzhen) Ltd., Shenzhen 518000, China

Corresponding author: Fangfang Yang (fangfyang2-c@my.cityu.edu.hk)

This research work was supported in part by the General Research Fund under Project CityU 11206417 and in part by the Research Grants Council Theme-based Research Scheme under Project T32-101/15-R.

ABSTRACT For most deep learning practitioners, recurrent networks are often used for sequence modeling. However, recent researches indicate that convolutional architectures may be used to optimize recurrent networks on some machine translation tasks. Problems here are which architecture we should use for a new sequence modeling. By integrating and systematically evaluating the general convolution and recurrent architecture used for sequence modeling, a convolution gated recurrent unit (CNN-GRU) network is proposed for the state-of-charge (SOC) estimation of lithium-ion batteries in this paper. Deep-learning models are well suited for SOC estimation because a battery management system is time-varying and non-linear. The CNN-GRU model is trained using data collected from the battery-discharging processes, such as the dynamic stress test and the federal urban driving schedule. The experimental results show that the proposed method can achieve higher estimation accuracy than two commonly used deep learning models (recurrent neural network and gated recurrent unit) and two traditional machine learning approaches (support vector machine and extreme learning machine) for SOC estimation of lithium-ion batteries.

INDEX TERMS State-of-charge estimation, convolutional gated recurrent unit, lithium-ion battery.

I. INTRODUCTION

With the continuous development of electric vehicles, lithium-ion batteries have become the mainstream of energy storage systems with their high energy, high power density and long life [1]. In order to ensure the safe and reliable operation of the battery, a battery management system is needed to monitor the lithium-ion battery to extend the battery life [2]. State-of-charge (SOC) estimation is one of the main contents of battery management system research. The SOC indicates the rate of the conserved energy of a battery to the overall conserved energy.

There are many mature SOC estimation methods that have been developed and applied, with Coulomb counting being the most common one [3]. In the Coulomb counting, the SOC value is obtained by integrating the current on the time axis. Although the Coulomb counting is easy to calculate, the measurement and computation errors are accumulated during the integration process, and the SOC initial value is not easily obtained. Therefore, in this method, the SOC estimation is easily deviated from the actual value. The voltage-based method first measures the open circuit voltage and then estimates the SOC by looking up the OCV-SOC (Open Circuit Voltage State) table. However, these methods cannot perform well for $LiFePO_4$ battery because of their non-stationary nature of the discharge process. Moreover, in voltage-based approaches, the battery should be turned on for a period of time, waiting for the open-circuit voltage to reach a stable state and measuring the open-circuit voltage value. Thus, it is not practical in applications where the battery cannot be opened from the electrical circuits [4], [5]. Impedance spectroscopy is another commonly used method for SOC estimation [6]. However, this approach is appropriate in laboratory tests but requires some additional measurements that are not practical applications [4], [5]. Kim *et al.* took temperature as one of the input variables of the first order

The associate editor coordinating the review of this manuscript and approving it for publication was Giambattista Gruosso.

dynamic circuit model. However, this model does not consider the effect of temperature on OCV-SOC [7]. Lee *et al.* developed a SOC estimation method which is on the basis of a fuzzy neural network [8]. This method was tested only in the case of constant current discharge. However, in actual situations, many batteries cannot be discharged in this state. These existing SOC estimation methods have their corresponding shortcomings that we have mentioned above. In this paper, we aim at developing a SOC estimation algorithm on the basis of deep-learning methods for following reasons. Firstly, a deep learning method can adaptively learn the network parameters and parameterize the model itself, thus reducing the workload of the researchers. Secondly, with the emergence of GPU, training time has been greatly reduced, big data can be transmitted to server in the cloud for off-line training. Thirdly, compared with traditional methods, it is more convenient to add temperature and other factors to train our models.

Advances in deep learning models, particularly recurrent neural networks (RNN) and long short term memory (LSTM), provide some useful ways to solve this problem [9]–[14]. Lipu *et al.* proposed a recurrent nonlinear autoregressive with exogenous input neural network model for SOC estimation of battery [15]. David *et al.* also applied this model to estimate the SOC of a battery pack of a Nissan Leaf electric vehicle [16]. The intuitive appeal of recurrent modelling is that the hidden state can act as a representation of everything that has been seen in the sequence so far. Basic RNN architectures are difficult to train [17], [18], so it is often replaced by more complex architectures, such as the LSTM [11], [19] and the GRU [20], [21]. As convolutional networks have been successfully used in sequences analysis for decades, other related works tried the combined architecture of RNN and CNN. These work include convolutional LSTM, which inserts a convolutional layer in LSTM to allow additional structures to be used in the recurrent layer and the Quasi-RNN model [22] which interlaces recurrent layers with convolutional layers. Although a number of evaluations of the RNN architecture of representative sequence modelling tasks have been conducted, we have not realized a thorough comparison of the convolution and rotation methods of sequence modelling. Yin *et al.* conducted a comparison between convolutional and recurrent networks for classification tasks [23]. In contrast, sequence modelling calls for architectures that can synthesize whole sequences, element by element. This is especially interesting given the recent success of the convolutional architecture mentioned in this area. Our work aims to compare recurrent architectures with and without general convolution for sequence modelling tasks.

In this study, a convolutional gated recurrent unit (CNN-GRU) is proposed for SOC estimation of lithium-ion batteries. The proposed CNN-GRU network is trained and tested using data collected from federal urban driven schedule (FUDS), dynamic stress test (DST), and US06 test under various temperatures. Experimental results show that the proposed network outperforms popular deep-learning methods like GRU and RNN and traditional machine learning methods such as extreme learning machine (ELM) and support vector machine (SVM) in terms of estimation errors like RMSE and MAE, with all models well trained on lithium-ion battery datasets. This paper is organized as follows. Section II describes the structure of convolution gated recurrent unit. Section III describes the data and experiment. Section IV shows the estimation procedures and results. Section V concludes.

II. CONVOLUTION GATED RECURRENT UNIT NEURAL NETWORKS

A. THEORY OF GRU

Cho *et al.* [20] proposes a GRU method that allows each recurrent unit to adaptively capture sequence dependencies over various lengths of time. The GRU has a gating structure which controls the flow of information inside the unit. In the following process, we denote x_t as current input, h_{t-1} is the output of previous stage.

1) Update gate.
The update gate z_t is computed by

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j. \tag{1}$$

An update gate z_t^j determines how much information the unit will update.

2) Reset gate.
The reset gate r_j^t is calculated by following equation,

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j, \tag{2}$$

but the weight matrix is different. Reset gates determine how much of the information hidden at the last moment needs to be forgotten. When the value is close to 0, it means that the j-th information of the previous moment is forgotten in the current memory content (explained in the following text), and close to 1 indicates that it will continue to be retained in the current memory content.

3) Determine the current memory content.
In addition to multiplying the weight matrix as above two gates, Hadamard product of the reset gate results r_t and h_{t-1} is calculated, i.e. the corresponding elements are multiplied. Because r_t is composed of vectors from 0 to 1, the significance of Hadamard product is to use the reset gate to decide how much of the hidden state of the content is to be forgotten in the current memory content. As described at the reset gate, the value close to 0 indicates that the information is forgotten and close to 1 preserves the information. Finally, the two parts of information are added into the tanh activation function, and the results are scaled down to - 1 to 1.

$$\widetilde{h}_t^j = \tanh(W_h x_t + U(r_t \cdot h_{t-1}))^j, \tag{3}$$

where r_t is a set of reset gates and $\cdot$ is an element-wise multiplication. When off (r_t^j close to 0), the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state.

4) Determine the information retained by the hidden layer at the current time.

Finally, the network needs to compute the h_t vector, which will retain the information of the current cell and pass it to the next cell. In this process, using the update gate, on the one hand, such as the first item of the formula, it determines how much information in the h_{t-1} of the previous moment needs to be retained in the hidden unit h_t at this time. On the other hand, as the second item of the formula, the information that needs to be forgotten is expressed by $(1-z_t^j)$ and updated by the corresponding content of the memory at this moment. In this way, the update door can determine what content needs to be collected in the current memory and the last hidden layer respectively.

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j\widetilde{h}_t^j, \tag{4}$$

See Figure 1 for the graphical illustration of the GRU cell.

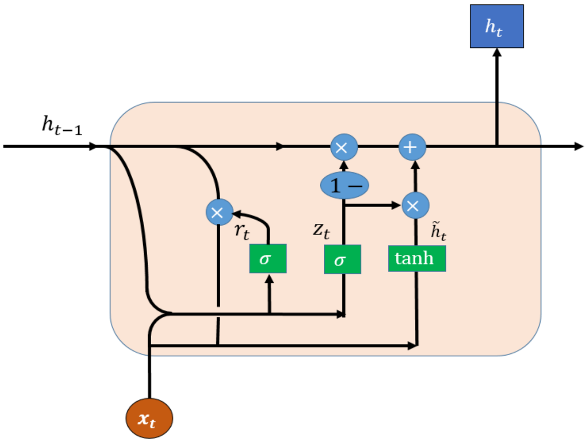


FIGURE 1. Structure of a GRU cell.

B. CONVOLUTION GRU

We now present the CNN-GRU network. Our architecture is inspired by convolutional architectures for sequential data, see [24]–[26] for references.

As we know, establishing a suitable estimation model is important to ensure satisfactory estimation accuracy. The SOC typically exhibit non-stationary and non-linear behavior, deeper architectures are well suited to make the network capture the non-linear properties. Therefore, this paper proposes a new deep learning model CNN-GRU for SOC estimation. Specifically, the CNN-GRU extracts features through a convolutional layer and performs time series prediction by stacking multiple GRU layers. Similar to other deep neural network models, the training procedure of CNN-GRU is implemented by using back propagation and gradient descent. The purpose of the training process is to reduce the root mean square error. In our applications, the self-adaptive learning rate optimization algorithm called AdaGrad is adopted. The optimization procedures of AdaGrad are as follows:

$$\alpha_e = \frac{\alpha_0}{\sqrt{\sum_{i=1}^{e} g_i^2 + \epsilon}}, \tag{5}$$

$$v_e = \mu v_{e-1} - \alpha_e g_e, \tag{6}$$

$$\beta_e = \beta_{e-1} + v_e, \tag{7}$$

where e is the epoch label; α_0 and α_e are initial learning rate and learning rate in epoch e, respectively; ϵ is a minimum value to ensure the denominator greater than 0; g_e denotes the gradient and μ denotes the momentum; β_e denotes parameters value and v_e is the updating value at the e th epoch.

CNN-GRU is based on following principles: the network will not leak to the future. Therefore, the CNN-GRU uses causal convolutions, which means an output at time t is only convolved with data at time t and time points before time t.

The key equation of CNN-GRU are shown below:

$$\widetilde{x}_t = \int x(t)g(s-t)dt, \tag{8}$$

$$z_t^j = \sigma(W_z\widetilde{x}_t + U_z h_{t-1})^j, \tag{9}$$

$$r_j^t = \sigma(W_r\widetilde{x}_t + U_r h_{t-1})^j, \tag{10}$$

$$\widetilde{h}_t^j = \tanh(W_h\widetilde{x}_t + U(r_t \cdot h_{t-1}))^j, \tag{11}$$

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j\widetilde{h}_t^j, \tag{12}$$

where $g(s-t)$ denotes the convolution function.

C. DISCUSSION

In this section, we discuss the effect of a one-dimensional convolution layer in this experiment.

Firstly, it is well known that one-dimensional convolution layer can preprocess data. By choosing the weight of convolution kernel and the width of window, different data features can be extracted to better serve as input of GRU layer.

Secondly, another function of the convolution layer is to make better use of the data before the current time point. From the mathematical definition of convolution, see following convolution equation

$$\widetilde{x}_t = \int x(t)g(s-t)dt,$$

it is equivalent to weighted average of current and past time points. In [27], average current and average voltage are also used as input parameters to estimate SOC. However, in this paper, we do not directly take average current and average voltage as input parameters. Instead, we let the convolution layer extract similar information adaptively and use it as input of GRU layer.

Thirdly, flexible receive field size. The CNN-GRU can change the size of its receive field in a number of ways. For example, using a larger expansion factor or increasing the filter size to stack more expanded convolution layers is a viable option. Therefore, CNN-GRU can easily adapt to different fields.

Finally, a convolution layer in the first place is similar to impose fast Fourier transform or wavelet transform on the

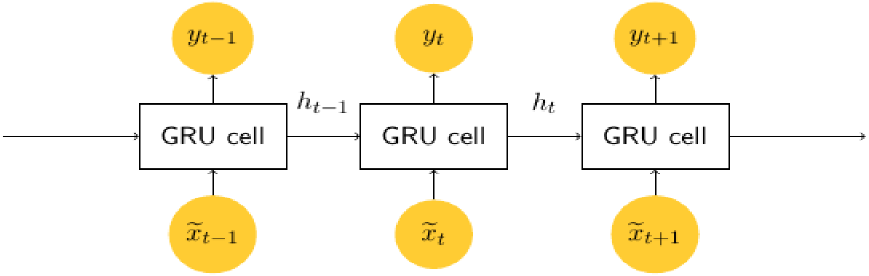


FIGURE 2. Structure of the GRU layer.

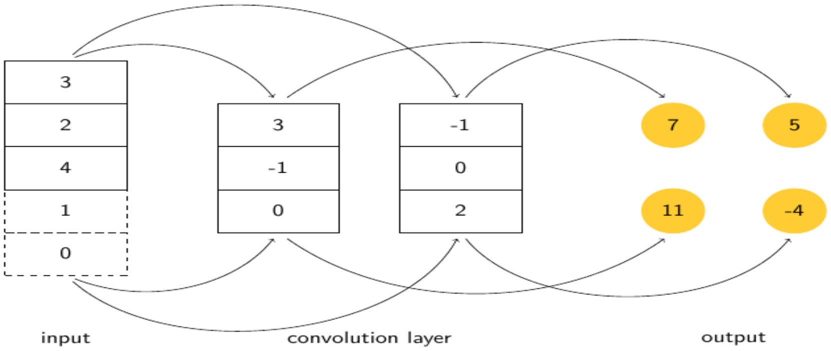


FIGURE 3. Structure of a 1-D convolution layer.

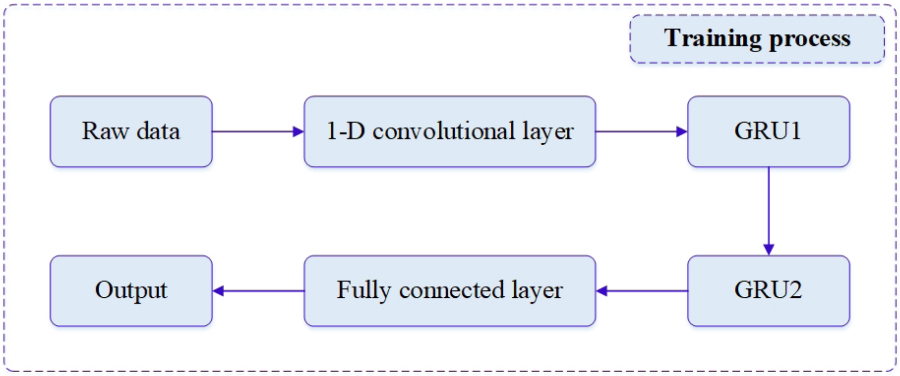


FIGURE 4. Flowchat of the training process.

raw data. However, the major difference here is that in fast Fourier transform or wavelet transform we use fixed kernel function and predetermined width of window. But in our experiment, the 1-D convolution layer is part of the whole neural network, which means all these parameters (weight and width of window) will be iteratively trained. Therefore, the network can self-adaptively select these parameters to reduce the estimation error.

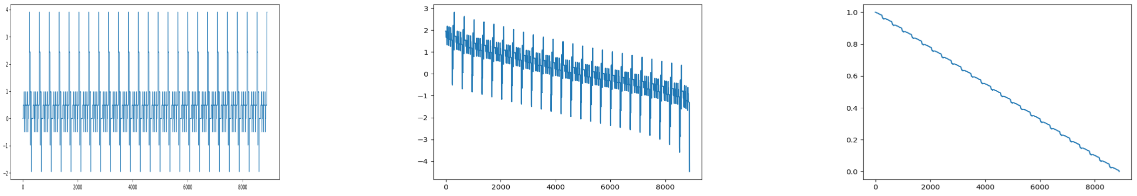
The structure of convolution layer is shown in Figure 3. The flowcharts of the proposed method for SOC estimation is given in Figure 4.

III. DATA AND EXPERIMENT

The battery tester consists of a battery testing system (arbin BT2000 tester) which loads and samples batteries, a host with Arbin MITs Pro on-line experimental control and data recording software, and a Pycharm data analysis software, as shown in Figure 7. The BAK 18650 battery (Li(NiCoMn)O_2), was used in the test, and the main parameters are in Table 1. The Federal Urban Driving Program (FUDS) and the dynamic stress test (DST) were conducted on the battery test bench at different temperatures for model evaluation, respectively, see following descriptions.

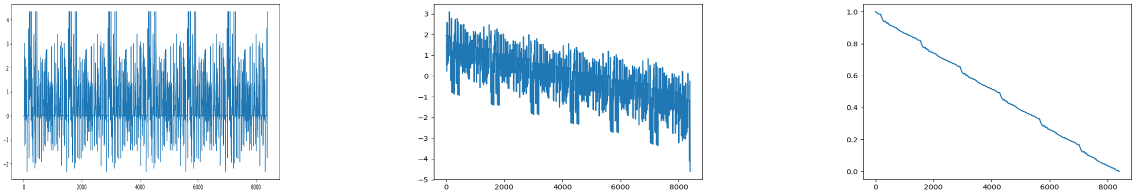
A. TRAINING AND TESTING DATA

Training data: the Federal Urban Driving Program (FUDS) is a dynamic performance test based on the time-speed curve of a vehicle in the automotive industry [28]. In laboratory tests, dynamic current sequences are transmitted over time-speed



(a) (b) (c)

FIGURE 5. Data from DST profile, (a) current of DST test; (b) voltage of DST test; (c) SOC of DST test.



(a) (b) (c)

FIGURE 6. Data from FUDS profile, (a) current of FUDS test; (b) voltage of FUDS test; (c) SOC of FUDS test.

TABLE 1. Specifications of the test samples.

Type	B18650CD Cathode Li$(\text{NiMnCo})\text{O}_2$ AnodeSi/C
Nominal capacity	1.3 A h
Nominal voltage	3.6 V
Charge voltage	4.2V
Discharge cut-off voltage	2.5V
End of charge current	0.01 C

curves to control battery charging or discharging and are used in battery performance testing. The current of FUDS causes the battery to discharge, thus SOC varies from 100% SOC at 3.6 V to empty state at 2 V. The FUDS test was conducted at 0°C, 10°C, 20°C, room temperature (26°C), 30°C, 40°C, 50°C. The SOC curve derived by Coulomb counting and measured current, voltage profile at room temperature are displayed in Figure 6.

Testing data: The dynamic electrical behavior of the battery was studied using DST profile. It was designed by the American Advanced Battery Alliance (USABC) to simulate a variable power discharge mechanism and represents the expected demand for electric vehicle batteries [28]. In our tests, the completed DST cycle was 360 seconds. During the discharge process, the DST runs from fully charged at 3.6 V to empty at 2 V continuously over five cycles in a discharge process. In our test, positive current corresponds to discharge processes and negative current corresponds to charge processes. The current and voltage curves at room temperature and the corresponding SOC curve are shown in Figure 6.

The structure of the input data of the network is [Current, Voltage, Temperature], where the temperature is controlled to oscillate around the preset temperature. The output data of the network is the estimation value of SOC.

Real values of SOC: The target value is calculated by Coulomb counting method. In this method, the SOC value is obtained by integrating the current on the time axis, see following equation.

$$\text{SOC} = \text{SOC}_0 - \frac{\int idt}{C_n}, \tag{13}$$

where SOC_0 is the initial value of SOC, i is the battery current, C_n renpresents the nominal capacity and t is the time.

B. EXPERIMENT

1) Experiment I: Hyper-parameter selection of the proposed method.
2) Experiment II: Comparative studies with deep-learning method such as RNN and GRU and with traditional machine learning approaches such as ELM [29] and SVM [30].
3) Experiment III: Testing robustness of the proposed method using different current discharging profile such as US06.
4) Experiment IV: SOC estimation at various temperature conditions using proposed method.

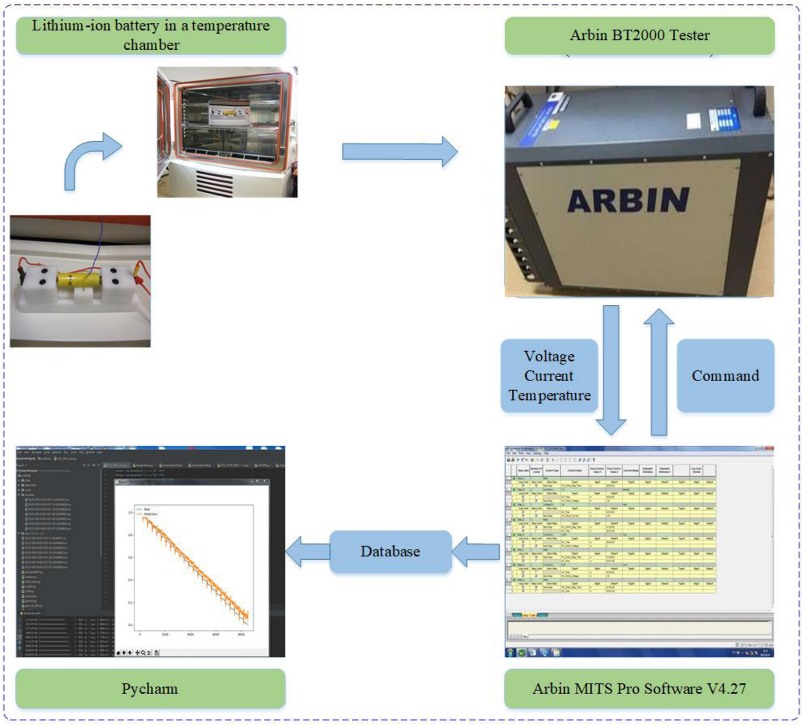

Lithium-ion battery in a temperature chamber
Arbin BT2000 Tester
ARBIN
Voltage
Current
Temperature
Command
Database
Pycharm
Arbin MITS Pro Software V4.27

FIGURE 7. Schematic of the battery test bench.

TABLE 2. Experiment I, hyper-parameters selection.

Model	RMSE	MAE
CNN(3*3)+GRU(150)+GRU(80)	0.0217	0.0198
CNN(6*3)+GRU(150)+GRU(80)	0.0184	0.0147
CNN(8*3)+GRU(150)+GRU(80)	0.0167	0.0133
CNN(8*3)+GRU(250)+GRU(80)	0.0185	0.0147
CNN(8*3)+GRU(100)+GRU(50)	0.0298	0.0288
CNN(8*3)+GRU(20)+GRU(10)	0.0492	0.0357
CNN(8*3)+GRU(150)	0.0176	0.0140

IV. TRAINING AND TESTING RESULTS

A. EXPERIMENT I: HYPER-PARAMETERS SELECTION

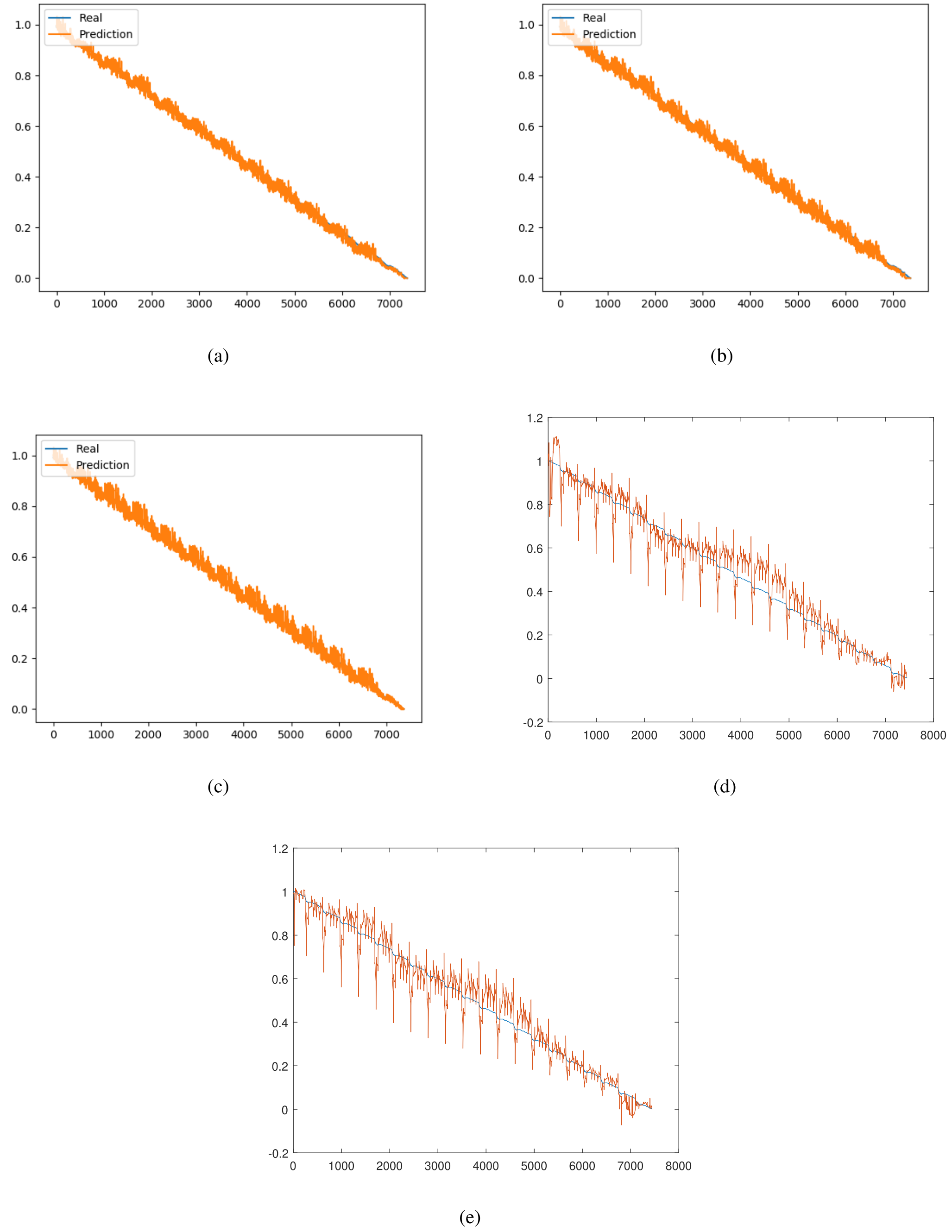
The architecture of the deep learning method has a great impact on network performance. In this section, several trials with different network structures are conducted to help determine proper hyper-parameters including the size of the convolution layer, the hidden layer, and hidden unit in each layer for the proposed network. The performance of the network is evaluated using mean absolute error (MAE) and root mean square error (RMSE), which are defined as,

$$\mathrm{MAE} = \frac{1}{T}\left(\sum_{t=1}^{T}|y_t - \hat{y}_t|\right), \tag{14}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}, \tag{15}$$

where $\hat{y}_t$ and y_t denote the estimated value and true value, respectively.

Some network specifications and results of the trials are summarized in Table 2. From the table, we see that although generally the network performance is improved with more hidden layers and more hidden nodes incorporated, the computational cost increases and more importantly, the risk of over-fitting increases. For example, first GRU layer with 250 nodes (Table 2, 4th row) yields larger estimation error than with 150 nodes (Table 2, row 3). Based on the evaluation results, two GRU layers are used. The structure of the network is shown in Figure 4, where 8 filters are adopted in the convolutional layer, followed by two stacked GRU layers, with 150 and 80 nodes, respectively. Finally, a fully connected layer is constructed for SOC regression. Other hyper-parameters, such as the epoch number, initial learning rate and momentum can be determined likewise. Specifically, 500 epoches are used, with initial learning rate and momentum set to 0.001 and 0.8, respectively. The learning rate is updated using AdaGrad [31] for faster convergence.



(a)

(b)

(c)

(d)

(e)

FIGURE 8. Experiment II, comparative studies with RNN, GRU, ELM and SVM; (a) CNN-GRU (b) GRU; (c) RNN; (d) ELM; (e) SVM.

Figure 8 (a) shows the estimated SOC for DST profile with the above specified hyper-parameters. The true SOC curve is calculated by Coulomb counting method, since the battery is discharged from 100% SOC and the current sensor is properly calibrated, the integration error is negligible.

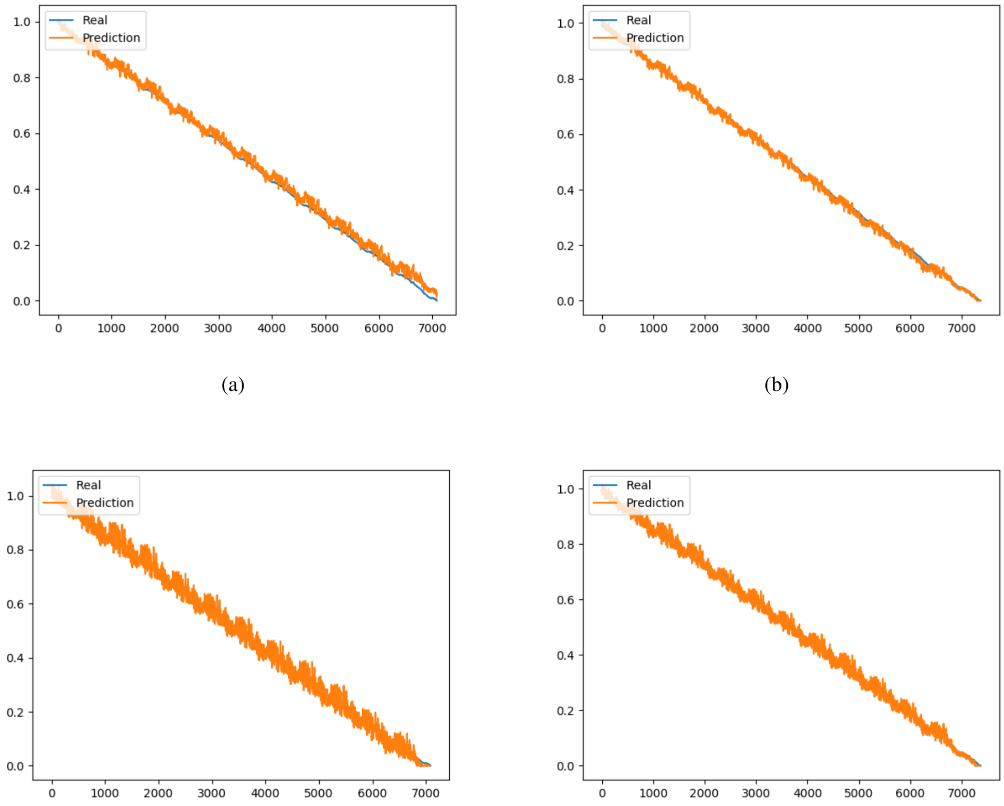
In conclusion, as with other deep learning methods, there are many hyper-parameters in the CNN-GRU network. Unfortunately, there is no mature way to properly select these hyper-parameters to ensure satisfactory results. In this paper, all of these hyper-parameters are manually and empirically selected by several attempts.

B. EXPERIMENT II: SOC ESTIMATION FOR FULLY CHARGED BATTERY UNDER ROOM TEMPERATURE USING DIFFERENT MODELS

In this section, the effectiveness and superiority of the proposed method are verified. In the experiment, the data from FUDS test and DST test are input into the CNN-GRU, GRU,

TABLE 3. Experiment II, comparative studies with RNN, GRU, ELM and SVM.

Method	RMSE	MAE	Average testing time (ms)
RNN	0.0184	0.0147	0.0411
GRU	0.0177	0.0147	0.0580
CNN-GRU	0.0167	0.0133	0.0435
ELM	0.0644	0.0480	0.0072
SVM	0.0583	0.0425	0.1757



(a)

(b)

(c)

(d)

FIGURE 9. Experiment III, (a) FUDS-US06-30°C; (b) FUDS-US06-40°C; (c) DST-US06-30°C; (d) DST-US06-40°C.

RNN, ELM, and SVM, respectively. The experiments are conducted on the same PC with CPU Intel(R) Core(TM) i7-2600 CPU, 3.40 GHz.

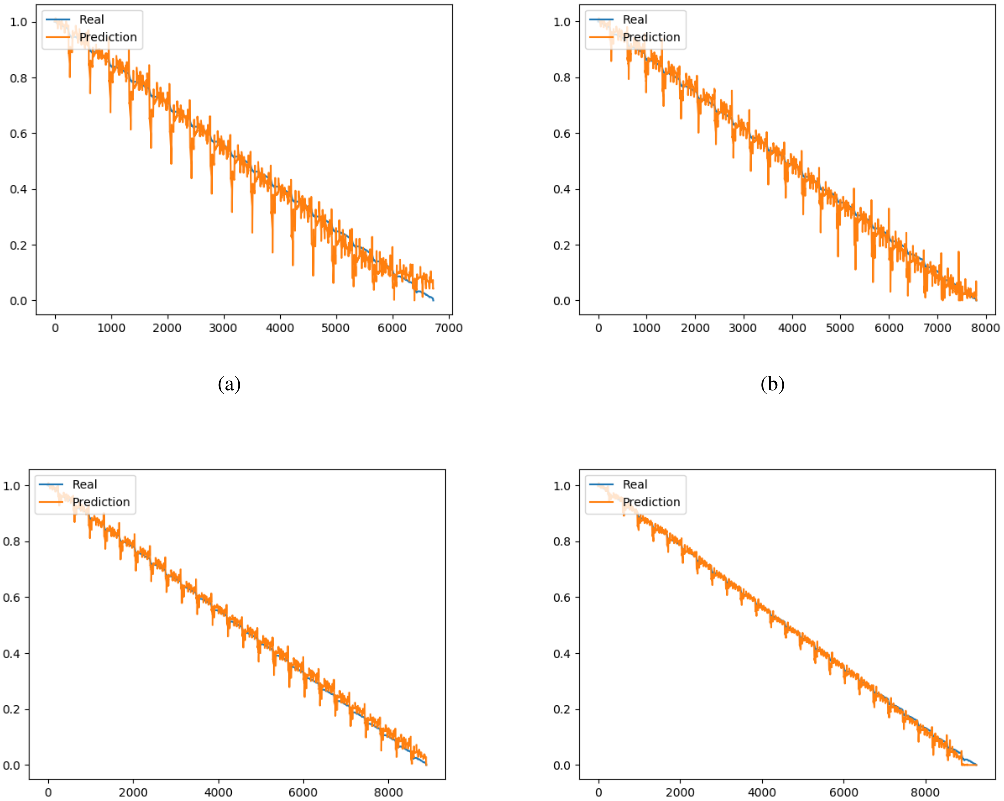
Figure 8 and Table 3 present the SOC estimation of the proposed network, where ELM, SVM, RNN, and GRU are implemented for comparison. The initial SOC is 100% since the battery is fully charged. The tracking results are shown in Figure 8. Table 3 summarizes all the results. The proposed method outperforms traditional machine learning approaches like ELM and SVM. The predicted values of ELM and SVM oscillate greatly around the real value, which results in large errors. The three deep learning methods can obtain satisfactory estimation results. The estimation accuracy of CNN-GRU is slightly better than the other two methods. The testing time of all the training models are less than 1 millisecond, which is fast enough for real-time applications. Specifically, the three deep-learning based models have very close computation time, the additional CNN layer has minor affects on the computational cost.

C. EXPERIMENT III: ROBUSTNESS OF THE PROPOSED METHOD AGAINST DIFFERENT LOADING PROFILES

In reality, battery loading varies greatly under different battery usages. For instance, electric vehicles driving in city and highway corresponds to different loading behaviours. In this section, the robustness of the proposed method against different loading profiles are considered by using data from different loading profiles from training and testing, respectively. For this purpose, additional experiments were done to collect data under a new loading profile - US06 [28] at

TABLE 4. Experiment III: Estimation results using CNN-GRU.

Data (training and testing)	RMSE	MAE
FUDS-US06-30°C	0.0198	0.0168
FUDS-US06-40°C	0.0098	0.0078
DST-US06-30°C	0.0211	0.0168
DST-US06-40°C	0.0130	0.0096



(a)

(b)

(c)

(d)

FIGURE 10. Experiment IV, (a) 0°C; (b) 10°C; (c) room temperature; (d) 40°C.

different temperatures. In the first experiment, we use data from FUDS test for training and data from US06 for testing. In the second experiment, we use data from DST for training and US06 for testing. Moreover, these experiments are repeated under 30°C, and 40°C, respectively. The network hyper-parameters are determined as in experiment I. The estimation results of all the tests are shown Figure 9, while the statistical results are tabulated in Table 4. For the all the tests, the maximum RMSE and MAE are within 2.2%, which is quite satisfying. Thus we conclude that the proposed method can be applied to various discharging patterns.

D. EXPERIMENT IV: SOC ESTIMATION AT VARIOUS TEMPERATURES

Ambient temperature has a great impact on the accuracy of SOC estimation. Traditionally, in model-based approaches, new battery model must be developed to take temperature into consideration. Unlike model-based approaches, which relies heavily on the accuracy of the underlying battery model and estimation algorithms, the deep learning method requires no battery model and can directly take temperature as an input factor to train the proposed neural network.

To see the influence of ambient temperature, in this section, we use FUDS data at 0°C, 10°C, 20°C, 30°C, 40°C and 50°C to train the proposed CNN-GRU neural network. The trained neural network, which maps the input vectors including current, voltage and temperature to the output SOC, is then tested with input data from various temperatures, including room temperature (around 26°C).

Figure 10 shows the SOC estimation results of the proposed method, while Table 5 lists the RMSEs and MAEs for all tests. Generally, the estimation performance under

TABLE 5. Experiment IV, estimation results under different temperatures using CNN-GRU.

Temperature	RMSE	MAE
0°C	0.0436	0.0304
10°C	0.0267	0.0195
26°C	0.0154	0.0126
40°C	0.0130	0.0095
50°C	0.0205	0.0164

different temperatures is satisfying, with maximum RMSE and MAE around 4%. Moreover, although no data at room temperature is used as the network training data, the SOC estimation performance at room temperature is satisfying, with RMSE and MAE within 2%. Finally, the worst performance happens at 0°C. The reason is that lithium-ion batteries have poor performance and more complex internal dynamics at low temperature, thus it is much harder for the proposed network to obtain useful features.

In short, by adding temperature into the network input, the proposed network can be easily extended to consider the influence of ambient temperature on battery SOC estimation. Overall, the proposed method can give relatively good SOC estimation at various temperatures. Moreover, the proposed network can present satisfying estimation results at an untrained temperature.

V. CONCLUSIONS AND DISCUSSION

Deep learning methods perform well in our experiments. Firstly, CNN-GRU can directly map battery measurement signals such as voltage, current, and temperature to SOC, avoiding complex model construction processes and computational inference algorithms such as Kalman filters used in traditional SOC estimation. Secondly, CNN-GRU can learn all parameters in the whole neural network by itself; releasing researchers from manual engineering and parameterizing the models themselves. In the traditional method, the algorithm can only be run in the field after fitting the parameters to the battery model and determining the time-consuming covariance matrix in the Kalman filter. Thirdly, the CNN-GRU network can learn to estimate the SOC under various ambient temperature conditions, which is very useful and easy to implement because in existing SOC estimation methods, different models or different lookup tables must be used for different ambient temperatures. Fourthly, the proposed method outperforms popular deep learning methods like RNN and GRU and traditional machine learning approaches like ELM and SVM in terms of estimation errors like RMSE and MAE. The computation time is fast enough for real-time applications. Finally, repeated use will affect the dynamic performance of the battery. In order to consider the effect of battery aging, we recommend that network parameters be updated regularly. In our experience, two months or even longer intervals are acceptable because the battery aging process is slow.

REFERENCES

[1] F. Yang, X. Song, G. Dong, and K.-L. Tsui, "A coulombic efficiency-based model for prognostics and health estimation of lithium-ion batteries," *Energy*, vol. 171, pp. 1173–1182, Mar. 2019.

[2] F. Yang, Y. Xing, D. Wang, and K.-L. Tsui, "A comparative study of three model-based algorithms for estimating state-of-charge of lithium-ion batteries under a new combined dynamic loading profile," *Appl. Energy*, vol. 164, pp. 387–399, Feb. 2016.

[3] K. S. Ng, C.-S. Moo, Y.-P. Chen, and Y.-C. Hsieh, "Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries," *Appl. Energy*, vol. 86, no. 9, pp. 1506–1511, Sep. 2009.

[4] Y. Xing, E. W. M. Ma, K. L. Tsui, and M. Pecht, "Battery management systems in electric and hybrid vehicles," *Energies*, vol. 4, no. 11, pp. 1840–1857, 2011.

[5] X. Chen, W. Shen, Z. Cao, and A. Kapoor, "Adaptive gain sliding mode observer for state of charge estimation based on combined battery equivalent circuit model," *Comput. Chem. Eng.*, vol. 64, pp. 114–123, May 2014.

[6] F. Yang, D. Wang, Y. Zhao, K.-L. Tsui, and S. J. Bae, "A study of the relationship between coulombic efficiency and capacity degradation of commercial lithium-ion batteries," *Energy*, vol. 145, pp. 486–495, Feb. 2018.

[7] J. Kim, S. Lee, and B. Cho, "The state of charge estimation employing empirical parameters measurements for various temperatures," in *Proc. IEEE 6th Int. Power Electron. Motion Control Conf.*, May 2009, pp. 939–944.

[8] Y.-S. Lee, W.-Y. Wang, and T.-Y. Kuo, "Soft computing for battery state-of-charge (BSOC) estimation in battery string systems," *IEEE Trans. Ind. Electron.*, vol. 55, no. 1, pp. 229–239, Jan. 2008.

[9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Aug. 2015, pp. 2625–2634.

[10] K. Cho, B. Van Merriénboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: https://arxiv.org/abs/1406.1078

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Aug. 2015, pp. 3128–3137.

[13] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMS," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 843–852.

[14] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[15] M. H. Lipu, A. Hussain, M. H. M. Saad, A. Ayob, and M. A. Hannan, "Improved recurrent NARX neural network model for state of charge estimation of lithium-ion battery using pso algorithm," in *Proc. IEEE Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, Apr. 2018, pp. 354–359.

[16] D. Jiménez-Bermejo, J. Fraile-Ardanuy, S. Castaño-Solis, J. Merino, and R. Álvaro-Hermana, "Using dynamic neural networks for battery state of charge estimation in electric vehicles," *Procedia Comput. Sci.*, vol. 130, pp. 533–540, Jan. 2018.

[17] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[18] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2013, pp. 1310–1318.

[19] F. Yang, X. Song, F. Xu, and K.-L. Tsui, "State-of-charge estimation of lithium-ion batteries via long short-term memory network," *IEEE Access*, vol. 7, pp. 53792–53799, 2019.

[20] K. Cho, B. Van Merriénboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: https://arxiv.org/abs/1409.1259

[21] F. Yang, W. Li, C. Li, and Q. Miao, "State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network," *Energy*, vol. 175, pp. 66–75, May 2019.

[22] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-recurrent neural networks," 2016, *arXiv:1611.01576*. [Online]. Available: https://arxiv.org/abs/1611.01576

[23] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, *arXiv:1702.01923*. [Online]. Available: https://arxiv.org/abs/1702.01923

[24] A. Van Den Oord, A. Graves, H. Zen, O. Vinyals, A. Senior, S. Dieleman, N. Kalchbrenner, K. Kavukcuoglu, and K. Simonyan, "Wavenet: A generative model for raw audio," in *Proc. SSW*, Sep. 2016, p. 125.

[25] D. Silver, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[26] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 1243–1252.

[27] 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. Elect. Power Energy Syst.*, vol. 62, pp. 783–791, Nov. 2014.

[28] G. Hunt, *USABC Electric Vehicle Battery Test Procedures Manual*. Washington, DC, USA: United States Department of Energy, 1996.

[29] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2. Jul. 2004, pp. 985–990.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[31] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*. [Online]. Available: https://arxiv.org/abs/1609.04747

ZHELIN HUANG received the Ph.D. degree in statistics from Chinese University of Hong Kong, Hong Kong, China, in 2018. He is currently a Postdoctoral Fellow with the School of Data Science, City University of Hong Kong. His research interests include time series analysis and prediction, spectrum analysis, and fault diagnostics.

FANGFANG YANG received the Ph.D. degree in systems engineering and engineering management from City University of Hong Kong in 2017. She is a Research Fellow with the School of Data Science, City University of Hong Kong. Her research interests include system health estimation, remaining useful life prediction, and deep learning.

FAN XU received the Ph.D. degree in mechanical engineering from Wuhan University, Wuhan, China, in 2017. He is currently pursuing the Postdoctoral Fellow with the School of Data Science, City University of Hong Kong. His research interests include machinery condition monitoring, intelligent fault diagnostics, and deep learning.

XIANGBAO SONG received the M.Phil. degree in electronic and computer engineering from Hong Kong University of Science and Technology, Hong Kong, P.R.C., in 2016. He is currently a Researcher with the Googol Tech, Shenzhen Research Institute. His research interests including robotics, motion planning, signal processing, prognostics and health management, and deep learning.

KWOK-LEUNG TSUI was a Professor with the School of Industrial and Systems Engineering, Georgia Institute of Technology. He was a recipient of the National Science Foundation Young Investigator Award. He is currently the Chair Professor of industrial engineering with the School of Data Science, City University of Hong Kong, and the Founder and the Director of Center for Systems Informatics Engineering. His current research interests include data mining, surveillance in healthcare and public health, prognostics and systems health management, calibration and validation of computer models, process control and monitoring, and robust design and Taguchi methods. He is a Fellow of the American Statistical Association, the American Society for Quality, and the International Society of Engineering Asset Management, and a U.S. representative to the ISO Technical Committee on Statistical Methods. He was the Chair of the INFORMS Section on quality, statistics, and reliability and the Founding Chair of the INFORMS Section on data mining.