SOC Estimation of Li-ion Batteries With Learning Rate-Optimized Deep Fully Convolutional Network

Hannan, Mahammad A.; How, Dickson N. T.; Lipu, Molla S. Hossain; Ker, Pin Jern; Dong, Zhao Yang; Manur, Muhamad; Blaabjerg, Frede

Published in:
I E E E Transactions on Power Electronics

DOI (link to publication from Publisher):
10.1109/TPEL.2020.3041876

Publication date:
2021

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Hannan, M. A., How, D. N. T., Lipu, M. S. H., Ker, P. J., Dong, Z. Y., Manur, M., & Blaabjerg, F. (2021). SOC Estimation of Li-ion Batteries With Learning Rate-Optimized Deep Fully Convolutional Network. *I E E E Transactions on Power Electronics*, *36*(7), 7349 - 7353. [9276459]. https://doi.org/10.1109/TPEL.2020.3041876

SOC Estimation of Li-ion Batteries with Learning Rate-Optimized Deep Fully Convolutional Network

M. A. Hannan, *Senior Member IEEE*, D. N. T. How, *Member IEEE*, M. S. Hossain Lipu,
P. J. Ker, *Member IEEE*, Z. Y. Dong, *Fellow, IEEE*, M. Mansur, F. Blaabjerg, *Fellow, IEEE*

***Abstract*—In this study, we train deep learning (DL) models to estimate the state-of-charge (SOC) of lithium-ion (Li-ion) battery directly from voltage, current, and battery temperature values. The deep fully convolutional network (FCN) model is proposed for its novel architecture with learning rate optimization strategies. The proposed model is capable of estimating SOC at constant and varying ambient temperature on different drive cycles without having to be re-trained. The model also outperformed other commonly used DL models such as the LSTM, GRU, and CNN on an open source Li-ion battery dataset. The model achieves 0.85% RMSE and 0.7% MAE at 25°C and 2.0% RMSE and 1.55% MAE at varying ambient temperature (-20 to 25°C).**

***Index Terms*—State-of-charge, convolutional neural network, CNN, FCN, deep learning, lithium-ion battery.**

I. INTRODUCTION

STATE-of-charge (SOC) is a crucial parameter in the battery management systems of electric vehicles (EV) that indicates the amount of charge left in its batteries [1]. Accurate SOC estimation is essential in ensuring the longevity and safety of the lithium-ion (Li-ion) batteries. The formal definition of SOC is the ratio of available capacity Q at time, t, to the nominal capacity Q_n [2] as given in Eq. (1).

$$\text{SOC} = \frac{Q(t)}{Q_n} \tag{1}$$

With existing sensor advancements, the SOC cannot be practically measured outside of the laboratory with controlled environment. However, since the SOC correlates well with a few observable quantities such as battery voltage, current, temperature, and so forth, these quantities are often used to estimate the SOC [3]. Presently, researchers are adopting machine learning (ML) methods in which the battery model is learned directly from the battery data instead of being hand-engineered in laboratories. Conventional ML methods such as Kalman filters [4], neural networks [5], fuzzy controllers [6], and various hybrid methods have been extensively explored throughout the literature. However, there has been a growing interest in using DL for battery modelling recently. Among notable DL related works include LSTM [7], GRU [8], DNN [9], CNN-LSTM [10] and so forth have proven to yield promising results. Most studies involve recurrent DL models which handles temporal data well. However, the computation cost of recurrent models is huge compared to its feedforward counterparts such as DNN or CNN. Additionally, recent advances in DL suggest that feedforward models can outperform recurrent models on a variety of benchmarks. This study proposes an optimised deep fully convolutional network (FCN) to estimate the SOC of a Li-ion battery. The following are main contributions of this work:

- The proposed FCN outperforms recurrent models on the training and test set when evaluated on novel drive cycles absent in the training set with least computation cost.
- Learning rate optimization strategies significantly improves the error rate of the unoptimized FCN model.

This study also incorporates various recent DL training strategies and best practices including the use of state-of-the art optimizer and activation function.

II. PROPOSED FCN MODEL

A. FCN Architecture

FCN is commonly used in computer vision tasks. However, it can be adapted to work with temporal data. In FCN, the convolution operation is applied across the time axis with a 1-dimensional kernel known as temporal convolution. The FCN used in this study is constructed by stacking multiple temporal convolution layers atop one another. Fig. 1 illustrates the proposed FCN architecture in which input matrix are the battery voltage, V_k, current, I_k , and temperature, T_k. Following that, four subsequent temporal convolutions are performed on the matrix by sliding each kernel across the time dimension as shown in Fig. 1. The convolution operations are performed with the order of the following kernel width size, w = [7, 5, 3, 1] and each convolution layer has number of kernels, n = [16, 32, 16, 1], respectively. Each convolution layer is followed by a batch normalization (BN) layer to accelerate training convergence and subsequently by an activation layer. We utilized a relatively new activation function known as the *Mish* activation [11]. The *Mish* activation has been shown to improve results in convolutional networks and provides a strong regularization effect on the model to reduce overfitting. We also include a global average pooling (GAP) layer instead of a fully connected layer for to reduce the number of parameters that leads to overfitting. To yield the output SOC, we run the resulting tensor from the GAP layer through a rectified linear unit (ReLU) clipped at a ceiling of 1.0.

B. Learning Rate Optimization

Learning rate (LR) is arguably the most important hyperparameter that has a major influence on the model performance. Too small a LR results in lengthy training time. Vice versa, an overly large LR causes the model to not converge.

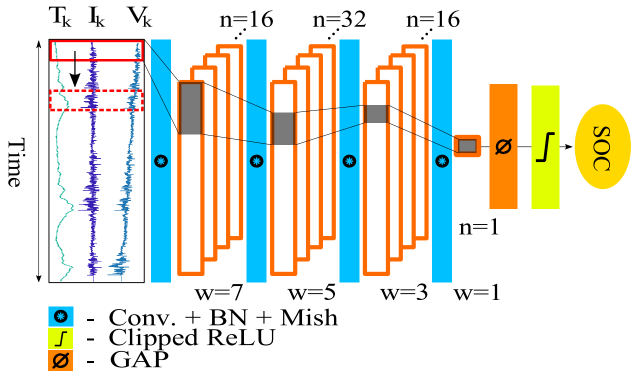


Fig. 1: Proposed FCN architecture.

This study adopts a search strategy to identify an optimal range of LR as proposed in [12]. During the searching, the training dataset is forward-passed through the model with exponentially increasing of LR. The above-mentioned search strategy is applied to minimize the loss function, $\mathcal{L}$ as expressed in Eq. (2). The mean absolute error (MAE) is chosen as the loss function with Ridge regularization (L2),

$$\mathcal{L} = \underbrace{\frac{1}{N}\sum_{k=1}^{N}(|SOC_k - SOC_k^*|)}_{Loss\ term} + \underbrace{\frac{\lambda}{2m}\sum_{j=1}^{n}\left\|w^j\right\|^2}_{L2\ regularization\ term} \qquad (2)$$

where, in the loss term, N is the total number of training samples, SOC_k is the estimated SOC by the model at timestep k, SOC_k^* is the ground truth SOC value at timestep, k. In the L2 regularization term, n is the number of layers, w^j is the weight matrix for layer j, m is the number of inputs, and λ is the regularization parameter. The outcome of the LR search is shown in Fig. 2. It can be observed that the loss function decreases at a different rate depending on the value of LR. The optimal LR value lies in the region where the loss decreases most rapidly. In our case, the optimal LR is within 10^{-4} to 10^{-2}.

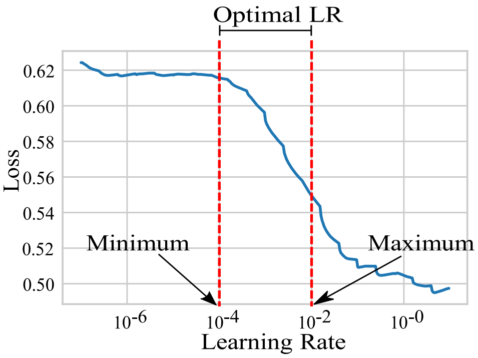


Fig. 2: Learning rate range finder.

Once the optimal range of LR is determined, a policy is applied to vary the LR during training in a cyclical fashion, as illustrated in Fig. 3. In this policy, the training starts with the minimum LR value (10^{-4}). As training progresses, the LR increases linearly until it reaches the maximum optimal LR (10^{-2}). The cycle is repeated until the model converges. Cyclically varying the LR increases the training speed as well as allows the models to avoid getting stuck on local minima. All models in this study are trained with gradient descent optimization algorithm known as Rectified Adam (RAdam) [13]. RAdam has been shown to be less sensitive to the selection of initial learning rate and has demonstrated improved generalization error.

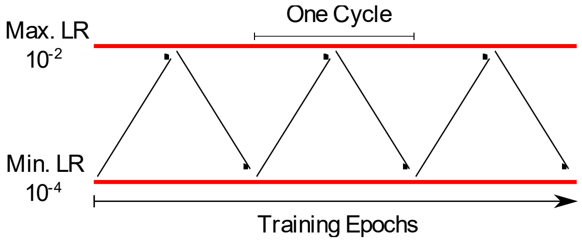


Fig. 3: Cyclical LR policy during training.

III. EXPERIMENTAL SETUP

A. Dataset

A Panasonic 18650PF lithium-ion battery cell with capacity of 2.9 Ah is employed in this research. The specification is presented in Table I [7]. Note that in this dataset, the discharging current is assigned as negative and charging current as positive. The dataset consists of 9 distinct drive cycles with over 100,000 timesteps of which seven are used in training and the remaining two as the test set. In this study, the training data was further split into 70/30 train/validation samples during training. Fig. 4 shows an unnormalized sample plot of the US06 dataset to illustrate the range of values for voltage, current, temperature and the available capacity of the battery. To ensure consistency and training stability, we re-sampled all data to 1Hz sample rate and normalized them in the range of 0 to 1.

B. Hyperparameters and Training

All models were trained on a dual 1080Ti GPU with (11Gb memory each) on an Ubuntu 18.04 32Gb RAM machine with Tensorflow 2.2.0 DL library. To reduce the amount of GPU memory usage, we casted the dataset into a half-precision (16-bit) floating-point format. This enables us to halve the amount of GPU memory usage and reduce computational complexity leading to speed up in training time. Halving the memory usage also allows to increase the batch size which further accelerates training. The batch size was kept at 1024 for all models. We have constructed a data window, W = 400 timesteps to train all models. The ground truth SOC_k^* is determined by using the Coulomb Counting formula with precisely calibrated sensor during the discharge process.

$$SOC_k^*(t) = SOC_k^*(t-1) + \frac{I(t)}{Q_n}\,\Delta t \qquad (3)$$

where Q_n is the nominal battery capacity, $SOC_k^*(t)$ is the present timestep SOC, $SOC_k^*(t-1)$ previous timestep SOC, and Δt is the time interval.

TABLE I: Panasonic 18650OF cell parameters [7].

Parameter	**Values**
Nominal open circuit voltage	3.6 V
Capacity	Min. 2.75 Ah / Typ. 2.9 Ah
Min/max voltage	2.5 V / 4.2 V
Mass/energy storage	48 g / 9.9 Wh
Minimum charging temperature	10 ˚C
Cycles to 80% capacity	500 (100% DOD, 25 ˚C)

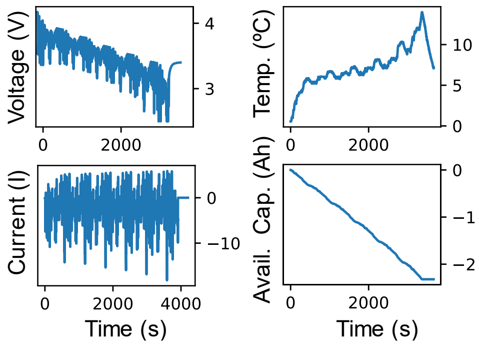


Fig. 4: Sample plot of the US06 test drive cycle sampled at ambient temperature, T = 10˚C.

In the model, we set λ = 0.001 and the loss function, L is optimized by the minibatch gradient descent with backpropagation. The weights and biases were updated using RAdam optimization as described in the previous subsection. To mitigate the effects of overfitting, the early stopping training scheme is adopted that halts training if the validation loss does not improve for 100 consecutive epochs. The maximum epoch was kept at 1000. The model with the lowest validation loss was selected as the best performing model. The performance of all models was evaluated with the Root Mean Squared Error (RMSE), and the Mean Absolute Error (MAE) given in the following equations:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(SOC_k - SOC_k^*)^2} \tag{4}$$

$$MAE = \frac{1}{N}\sum_{k=1}^{N}(|SOC_k - SOC_k^*|) \tag{5}$$

IV. EXPERIMENTAL VALIDATION

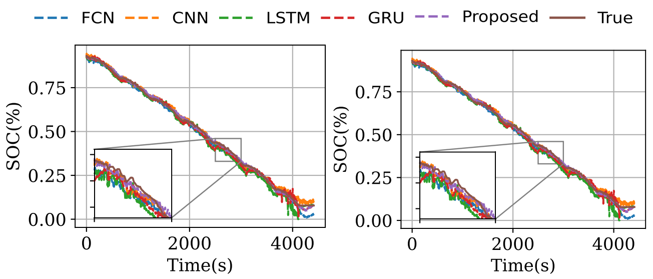
Results obtained by the proposed model is compared with other commonly used DL models such as the LSTM, GRU, and CNN. The LSTM and GRU models consist of one hidden layer with 32 and 36 units respectively. The CNN model used consists of a single temporal convolution layer with 22 filters and a kernel size of 5 followed by a max pooling layer with pool size 2. The number of hidden units and filters were selected such that all models consist of approximately the same number of parameters for a fair comparison.

A. Estimation at Constant Ambient Temperature

In this section, we trained and tested all models only on the drive cycles taken at room temperature (25˚C). Training data contains of drive cycles: Cycle 1,2,3,4, NN, UDDS, LA92 and testing data contains of drive cycles: US06 and HWFT. We observed (in Table III) that FCN outperformed all other models even without any optimization in achieving low RMSE and MAE during the testing phase. This is evident that the architecture of the FCN contributes to low error test error. In our proposed model, we performed LR optimization on the FCN model that has contributed to a significant error reduction in comparison to only FCN. The results prove that the LR optimization plays an important role in reducing test error. In summary, the proposed model is superior to other models under constant ambient temperature setting with respect to generalization capacity in obtaining the lowest error rates under testing phase. Fig. 5 illustrates the SOC estimation plot for all models trained under fixed ambient temperature.

TABLE II: Computation cost for all models.

Model	**Parameters**	**FLOPs**	**Run-time(s)**
Proposed	4643	9226	0.001361
FCN	4643	9226	0.001342
GRU	4543	17772	0.001478
LSTM	4711	18938	0.001368
CNN	4055	20067	0.001028



(a) US06 drive cycle.

(b) HWFT drive cycle.

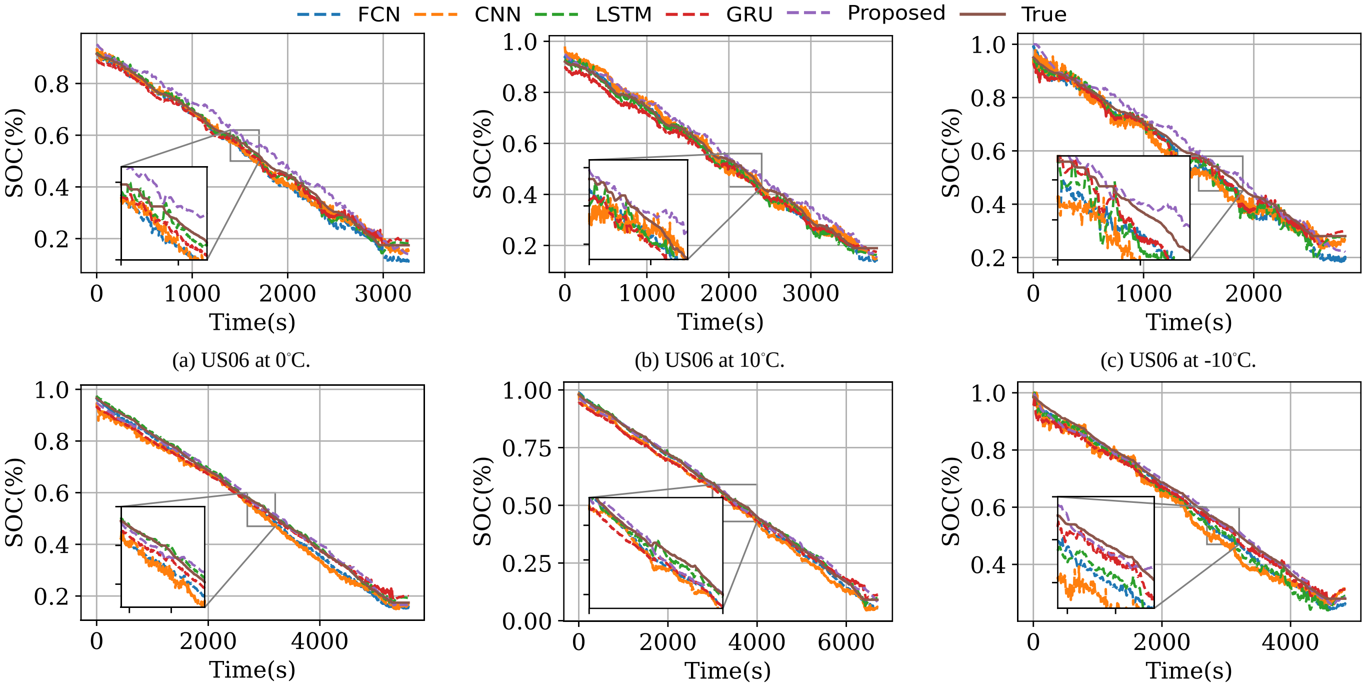
Fig. 5: Estimation at 25˚C ambient temperature.

B. Estimation at Variable Ambient Temperature

In this section, training and testing of all models on the drive cycles are taken at temperature ranging from -20˚C to 40˚C. Training data consists of drive cycles: Cycle 1,2,3,4, NN, UDDS, LA92 and testing data consists of drive cycles: US06 and HWFT, respectively. Table III shows the results on the training and test set of fixed ambient and variable ambient temperatures. Under the variable ambient temperature settings, we observe a similar pattern as before. FCN with no LR optimization already outperforms other models on the test error. The error rate on of the FCN decreases further (-0.45% RMSE, -0.31% MAE) with LR optimization. Fig. 6 illustrates the SOC estimation plots for all models at various ambient temperature values. However, upon careful observation, we note that the error of the proposed model on the train set increases despite a decrease on the test set. This phenomenon can be attributed to the "regularization effect" which is common in machine learning where the error on the train set increases and vice-versa on the test set. In many machine learning problems, regularization is deliberately added using various techniques such as Dropout to avoid overfitting on the training set. In our experiment setup, the regularization effect is already present by training the models with the cyclical LR policy. This effect is unintended but desirable since it improves model performance by not overfitting the training set. Regularization effect by training models with cyclical LR policy has also been documented in the seminal work by L. N. Smith in [12]. Upon observation in Fig. 5 and Fig. 6, the proposed model can match the trend of the true value in the most drive cycles despite the trend is not included in training. Thus, it is concluded that the model is capable to follow the flat trend of the true SOC value. However, more data including trend in the training and testing will provide maximum efficiency of matching true and proposed SOC value.

C. Computation Cost

To evaluate the computation cost and performance, two established metrics are utilized such as floating-point operations per second (FLOPs) and run-time for all models. FLOPs measure the number of operations per second for a trained DL model which is a good indicator to show the complexity of a model. Run-time is the time that takes to run one forward-pass through a model. The runtime performance is dependent on the hardware the model is run on. In this study, all models were tested on a single GTX1080Ti GPU. Fig. 7 illustrates the computational cost comparison across all models with respect to the test set error. One advantage of using convolutional models is that they are less computationally intensive compared to recurrent models. As shown in Fig. 7, the proposed model is computationally efficient compared to the LSTM and GRU while still scoring the lowest on the RMSE and MAE.



(a) US06 at 0°C.

(b) US06 at 10°C.

(c) US06 at -10°C.

(d) HWFT at 0°C.

(e) HWFT at 10°C.

(f) HWFT at -10°C.

Fig. 6: Estimation at various varying ambient temperatures.

TABLE III: Performance evaluation at fixed and varying ambient temperatures

		Fixed ambient temperature (25°C)						Variable ambient temperature (-20 to 25°C)					
		Training Error (%)			Test Error (%)			Training Error (%)			Test Error (%)		
Model	Params.	RMSE	MAE	MAX	RMSE	MAE	MAX	RMSE	MAE	MAX	RMSE	MAE	MAX
Proposed	4643	0.57	0.45	2.41	**0.85**	**0.70**	**2.96**	2.27	1.98	11.23	**2.00**	**1.55**	**7.63**
FCN	4643	0.59	0.45	3.27	1.48	1.11	7.10	1.31	0.97	9.63	2.45	1.86	11.62
GRU	4543	0.67	0.54	4.24	1.58	1.33	6.24	2.64	2.13	18.15	3.25	2.72	16.62
LSTM	4839	0.59	0.46	5.04	1.70	1.40	10.41	2.34	1.53	24.47	3.61	2.32	20.05
CNN	4759	0.84	0.63	7.84	2.17	1.88	5.63	2.28	1.65	17.31	3.88	2.76	18.06

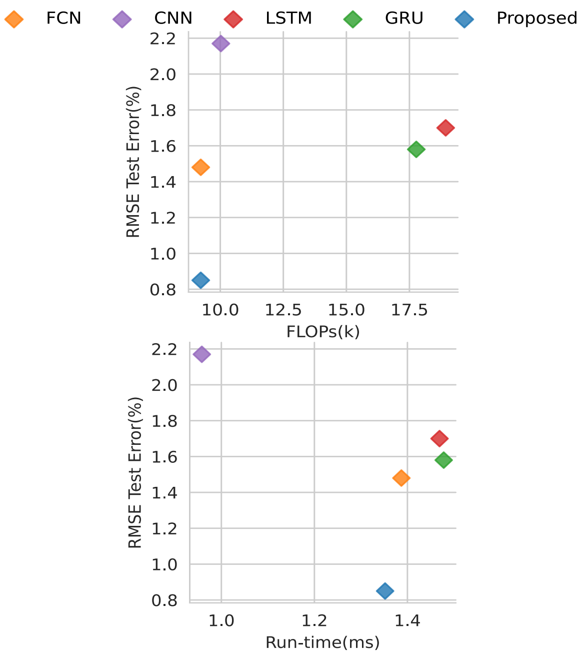


Fig. 7: Test set error versus computation cost for all models.

V. CONCLUSION & FUTURE WORKS

We proposed a novel DL architecture capable of accurately estimating the SOC in a fixed and variable ambient temperature setting. The proposed model outperformed conventional DL models such as the LSTM, GRU, and CNN by scoring the lowest RMSE, MAE, and MAX metrics. We demonstrated that optimization in the LR leads to improvement in estimation error and generalization capability. Furthermore, we show that the proposed model is computationally efficient with the least FLOPs and run-time speed. In this study, the proposed feedforward deep FCN with appropriate hyperparameters combination and learning rate optimization can outperform conventional DL models on the SOC estimation task. However, the superiority of the proposed model has not been validated with other battery types and will be subject of our upcoming works. Also, more data inclusion is suggested in the training algorithm on the parameters, loss function and trend to improve the error rate and the trend of SOC estimation in the future works.

VI. ACKNOWLEDGMENT

This work is supported by the Ministry of Higher Education, Malaysia under the LRGS grant: 20190101LRGS.

REFERENCES

[1] M. Hannan, M. H. Lipu, A. Hussain, P. J. Ker, T. Mahlia, M. Mansor, A. Ayob, M. H. Saad, and Z. Dong, “toward enhanced state of charge estimation of lithium-ion batteries using optimized machine learning techniques,” *Scientific reports*, vol. 10, no. 1, pp. 1–15, 2020.

[2] J.-N. Shen, J.-J. Shen, Y.-J. He, and Z.-F. Ma, “Accurate state of charge estimation with model mismatch for li-ion batteries: a joint moving horizon estimation approach,” *IEEE Transactions on Power Electronics*, vol. 34, no. 5, pp. 4329–4342, 2018.

[3] Q. Ouyang, J. Chen, and J. Zheng, “State-of-charge observer design for batteries with online model parameter identification: A robust approach,” *IEEE Transactions on Power Electronics*, vol. 35, no. 6, pp. 5820–5831, 2019.

[4] A. El Mejdoubi, A. Oukaour, H. Chaoui, H. Gualous, J. Sabor, and Y. Slamani, “State-of-charge and state-of-health lithium-ion batteries diagnosis according to surface temperature variation,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2391–2402, 2015.

[5] H. Dai, G. Zhao, M. Lin, J. Wu, and G. Zheng, “A novel estimation method for the state of health of lithium-ion battery using prior knowledge-based neural network and markov chain,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 10, pp. 7706–7716, 2018.

[6] A. Fotouhi, D. J. Auger, K. Propp, and S. Longo, “Lithium–sulfur battery state-of-charge observability analysis and estimation,” *IEEE Transactions on Power Electronics*, vol. 33, no. 7, pp. 5847–5859, 2017.

[7] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, “Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6730–6739, 2017.

[8] B. Xiao, Y. Liu, and B. Xiao, “Accurate state-of-charge estimation approach for lithium-ion batteries by gated recurrent unit with ensemble optimizer,” *IEEE Access*, vol. 7, pp. 54192–54202, 2019.

[9] D. N. How, M. Hannan, M. H. Lipu, K. S. Sahari, P. J. Ker, and K. M. Muttaqi, “State-of-charge estimation of li-ion battery in electric vehicles: A deep neural network approach,” *IEEE Transactions on Industry Applications*, 2020.

[10] X. Song, F. Yang, D. Wang, and K.-L. Tsui, “Combined cnn-lstm network for state-of-charge estimation of lithium-ion batteries,” *IEEE Access*, vol. 7, pp. 88894–88902, 2019.

[11] D. Misra, “Mish: A self regularized non-monotonic neural activation function,” *arXiv preprint arXiv:1908.08681*, 2019.

[12] L. N. Smith, “Cyclical learning rates for training neural networks,” in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472. IEEE, 2017.

[13] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, “On the variance of the adaptive learning rate and beyond,” *arXiv preprint arXiv:1908.03265*, 2019.