ASEAN Engineering Journal

Full Paper

ACCURATE STATE OF CHARGE ESTIMATION OF LITHIUM-ION BATTERY USING RECURRENT AND NON-RECURRENT NEURAL NETWORKS FOR WLTP DRIVING PROFILES

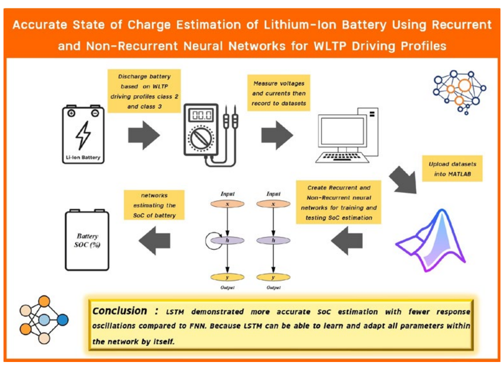
Article history
Received
03 July 2023
Received in revised form
21 October 2023
Accepted
12 December 2023
Published online
30 November 2024

Akkarawat Praisan*, Sompob Polmai, Supat Kittiratsatcha.

Department of Electrical Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand.

*Corresponding author
65016179@kmitl.ac.th

Graphical abstract



Abstract

Estimating the state of charge (SoC) of a battery is essential to maximize its performance and ensure reliable operation and battery life. Nowadays, many countries are increasingly adopting electric vehicles (EVs) with lithium-ion batteries due to their high specific energy and long service life. This paper presents a method for estimating the state of charge of lithium-ion batteries using artificial neural networks, specifically the Feedforward Neural Network (FNN) and Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM), through a data-driven approach. The training and testing of the networks are conducted using recorded datasets of the battery, based on the WLTP driving profiles class 2 and class 3. These driving profiles are specifically designed for testing electric vehicles, thereby enhancing the realism of the state of charge estimation by the network. In terms of the analytical aspect, the FNN was able to train the network faster due to its simpler structure, requiring less computation. On the other hand, the LSTM demonstrated more accurate SoC estimation with fewer response oscillations, thanks to its ability to learn and adapt network parameters internally.

Keywords: Battery, State of charge (SoC), Neural networks, Electric vehicle, WLTP, Data-driven

1.0 INTRODUCTION

Climate and weather scientists track human activities and influences. including the combustion of industrial fuels and the use of cars on the road, which has led to the release of large amounts of carbon dioxide and greenhouse gases into the atmosphere. This disrupts the global carbon cycle and leads to global warming [1]. As a result, many countries have begun to switch from using fuel-powered vehicles to battery-powered electric vehicles. The most popular battery used in electric vehicles is the lithium-ion battery the advantage of this type of battery is less pollution and short charging time. It has a long service life and more importantly, has a higher specific energy than other batteries.

The remaining energy or electric charge within the battery that lithium-ion batteries are stored and supply to the load is called "State of Charge" or SoC, which is critical to the reliable operation of an electric vehicle (EV) because the SoC directly indicates the distance an electric vehicle can drive and is necessary for a battery management system (BMS) [2]. However, the battery is a nonlinear function and has changed which is uncertain. It varies according to temperature and charge or discharge currents [3], and there exists no direct way to measure the SoC, therefore, an estimation method of the SoC must be used instead.

The Coulomb counting method involves measuring the battery current and integrating it over time to determine the state of charge (SoC) of the battery. However, this method may introduce errors due to sensor inaccuracies or drift. On the other hand, the OCV (Open Circuit Voltage) method determines SoC by measuring the battery's voltage when it is at rest, but it requires a significant resting period. Consequently, the OCV method is not suitable for real-time SoC estimation in electric vehicles, as accurate and timely SoC estimation is crucial for estimating the maximum driving distance with precision.

The technique currently used to estimate SoC values is based on a machine learning technique called Artificial Neural Network (ANN). ANNs can be divided into several types. This paper will demonstrate the SoC estimation of two different types of neural networks which are Feedforward Neural Networks (FNN) and Recurrent Neural Networks (RNN) with Long Short-term Memory (LSTM) cells. This type of SoC estimation is called data-driven methodology [4], which uses the data recorded from the battery in various ways to be used in training and testing the network to be able to estimate the SoC accurately.

2.0 METHODOLOGY

In this section, the two types of neural networks used to estimate SoC in this research which are FNN, and RNN-LSTM, are presented. Different of both types of neural networks are described for further comparison.

2.1 Feedforward Neural Network

Feedforward Neural Network (FNN) is the basic neural network [5]. Its primary operation involves forwarding the data through the network, allowing computations to determine the network's solution. FNN consists of an input layer, a hidden layer, and an output layer. Commonly, an input layer and an output layer are single, but a hidden layer can have multiple layers. Figure 1 presents the structure of an FNN when used to estimate SoC.

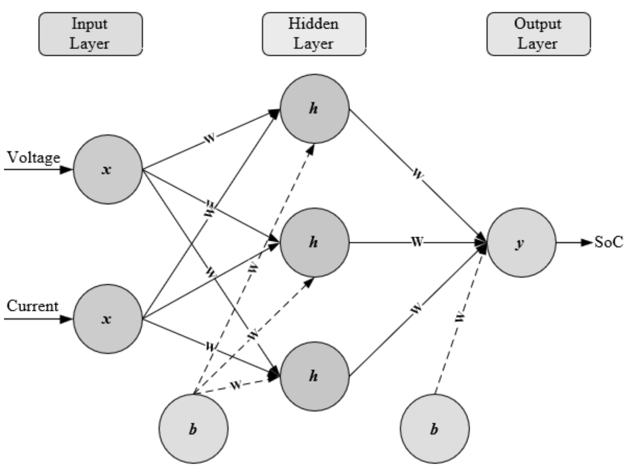


Figure 1 The structure of FNN

Two more things are hidden in the hidden layers, Weights, and Biases. Every hidden layer has weights as the weight of the data. As a result, all neurons have different outputs. The biases are connected so that the network is computationally flexible and works with all data formats. Equation (1) can be presented as SoC estimation as follows:

$$SoC = xw_y + b_y \tag{1}$$

Where x are the input vectors of the network consisting of the voltage and current of the battery, w_y and b_y are the output weight and the bias of the network, respectively. Processing is initiated when an input vector is fed into the network. The network then computes and produces an estimated SoC value at that specific timestep. It's important to note that the network is designed to work specifically for the output values it has been trained on. In other words, the network's purpose is to accurately determine the desired output values based on the input data it has been trained with. Due to the simplicity of the model structure and operation, FNNs are well suited to implementing problems that are not complicated.

2.2 Recurrent Neural Network

ANNs that are reconnected are called RNNs, where RNNs are a form of ANNs [6] that focuses on pattern recognition in sequential datasets [7]. It is also useful for all types of time series data. RNNs use output data from a previous timestep, t-1, as input, along with input data in the current timestep, t, to improve network computational efficiency. The structure of an RNN is shown in Figure 2 when using RNN to estimate SoC.

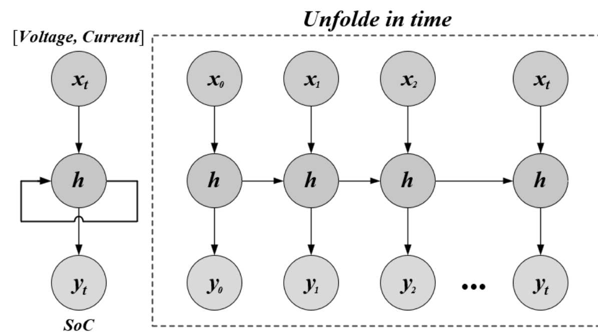


Figure 2 The RNN structure

The input vector that is fed into the network consists of x_t = [V_t, I_t] as the input vector, where V_t and I_t are represented by the voltage and current of the battery measured at the timestep t, respectively. The output SoC estimated by RNN at timestep t-1 is also fed to the network at timestep t, to increase the efficiency of network estimation. The SoC remaining in the battery also depends on the consumed and previous remaining energy. The function of RNN can be described as the following equation:

$$a_t = h_{(t\text{-}1)} w_y + x w_x + b \tag{2}$$

$$h_t = \boldsymbol{tanh}(a_t) \tag{3}$$

$$y_t = h_t w_y + b \tag{4}$$

where w_x, w_y, and b are input layer weights, output layer weights, and biases, respectively, h_t and $h_{t\text{-}1}$ are hidden layers at current timestep t and previous timestep t-1, y_t is the network output or SoC estimated. The problem with RNNs is the exploding and vanishing gradients while backpropagation. The RNN will backpropagation the errors to update w and b the networks for suitability and flexibility. A high value of the network causes an exploding gradient. Conversely, a low value of error w causes a vanishing gradient. The solution is to add a gradient clipping function in case of an exploding gradient. In the case of vanishing gradient, it needs to be solved by using Gate Recurrent Unit (GRU) or Long Short-Term Memory (LSTM) cells. This paper will use LSTM cells to estimate the SoC of lithium-ion batteries.

2.3 Long Short-Term Memory

The LSTM cell was developed from RNN to solve the problem of exploding gradient and vanishing gradient. The key component that makes LSTM different from RNN is adding cell state (Memory) to store data within the network [8]. Another addition is that the gates consist of an input gate, a forget, and an output gate. The architecture of LSTM is shown in Figure 3.

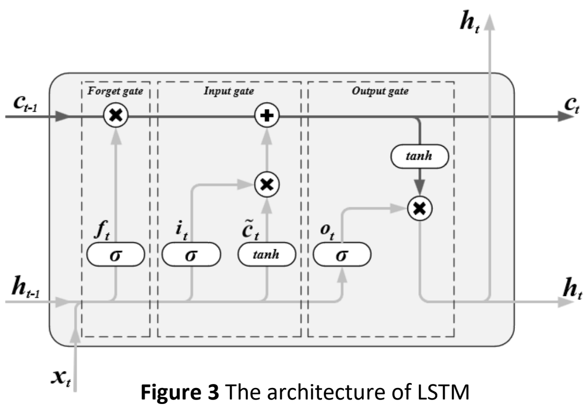


Figure 3 The architecture of LSTM

The current input vector x_t = [V_t , I_t] and the output from the previous timestep, h_{t-1}, are fed into the LSTM cell, along with the cell state of the previous timestep, c_{t-1}. Then, the data was passed through a forget gate to remove unnecessary old data in the cell state, the input gate adds new data and parameters into memory, and the data in memory after passing through the input gate and forget gate is computed together with input through the output gate to find cell output, h_t = SoC at every interval. The output and cell state at the current timestep, t, are sent to the next timestep, $t+1$. The LSTM structure can be described by the following equation:

$$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + b_i) \tag{5}$$

$$f_t = \sigma(w_{xf}x_t + w_{hf}h_{t-1} + b_f) \tag{6}$$

$$\tilde{c}_t = f_t c_{t-1} + i_t \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \tag{7}$$

$$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + b_o) \tag{8}$$

$$h_t = o_t \tanh(c_t) \tag{9}$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \tag{10}$$

Where σ represented by the sigmoid function, i is the input gate, f is the forget gate, o is the output gate and $\tilde{c}$ is the cell state or memory used to store network data and parameters. The output cell, h, is calculated at each interval and memory, c, is the storage at each interval. Therefore, h and c will be passed to the next timestep, $t+1$, to help the next cell calculate its output better. The output that the LSTM cell can calculate at each time step h_t, is the SoC value at each timestep, t, which can be obtained from the following equation:

$$SoC_t = w_y h_t + b_y \tag{11}$$

When using LSTM, exploding gradient, and vanishing gradient problems are avoided, because LSTM does not need Backpropagation. After all, it can learn and adjust all the parameters within the network by itself during training.

2.4 Experimental Setup

In this research, an experimental model of the Lithium-ion battery, namely LG MJ1, was used. LG MJ1 is known for being a high-capacity 18650 battery. It's an INR chemistry battery composed of Lithium Nickel Manganese (LiNiMnCoO2) [9] with a maximum capacity of 3.25 Ah, and a maximum current of 10 A, which is described in Table 1. This research was performed using one LG MJ1 battery recorded in a thermal chamber maintained at 25˚C.

Table 1 LG 18650 MJ1 cell Parameters

Items	Specification
Energy	3.25 Ah
Nominal Voltage	3.7 V
Min / Max Voltage	2.5V / 4.2 V
Max Charge Voltage	4.2 ± 0.05 V
Charge Current	Min. 3 A (0.5C) Max. 3.25 A (1C)
Max Discharge Current	10 A
Temperature	Charge: 0 ~ 45˚C Discharge: -20 ~ 60˚C

The dataset used in this test is based on the "Worldwide Harmonized Light Vehicles Test Procedure: WLTP" [10], the organization preparing to replace the NEDC. The testing of Electric Vehicles in Europe requires to change from NEDC to WLTP [11] because this new type of testing will take the test pattern from real road situations as a component in the test as well to get the value as close to reality as possible.

In general, the power consumption of vehicles is typically between 40-100 watts per kilogram of vehicle weight. Based on this classification, cars commonly fall under Class 3, whereas Class 2 predominantly comprises buses and trucks. Consequently, this research will bring the driving profile data WLTP Class 2 and Class 3 were used to train and test the estimation of SoC for FNN and LSTM networks using MATLAB. Hence, each network test was divided into 4 test cases as shown in Table 2.

Table 2 The test cases of SoC estimation with WLTP driving profile datasets

Cases	Train	Test
1	Class 2	Class 2
2	Class 3	Class 3
3	Class 2	Class 3
4	Class 3	Class 2

The datasets were recorded at the temperature of 25˚C

Firstly, the data recording begins by setting the thermal chamber to 25˚C, then discharging the battery based on the WLTP driving profiles and measuring the battery voltage and current until the battery is completely discharged. The data used to train and test the networks consists of voltage and current as input data. Integrating the currents over time helps determine the battery's state of charge (SoC). The obtained SoC will be used as a target to test the networks for estimation efficiency. The SoC of the battery can be indicated in equation (12),

$$SoC(t) = SoC(t_0) + \frac{\int_{t_0}^{t} i(t)dt}{Qrated} \tag{12}$$

where $SoC(t)$ and $i(t)$ are the state of charges and currents of the battery at each timestep, t, respectively, while $SoC(t_0)$ is the initial state of charge (usually set to 100%), and Q_{rated} is the battery's rated capacity. In addition, input voltage and current data of the WLTP Class 2 and Class 3 driving profiles are normalized to have a minimum value of 0 and a maximum value of 1 before being fed into the network. Since the battery is only tested at an ambient temperature of 25˚C, the only data used as network inputs are voltage and current. As shown in Figure 4

(a)

(b)

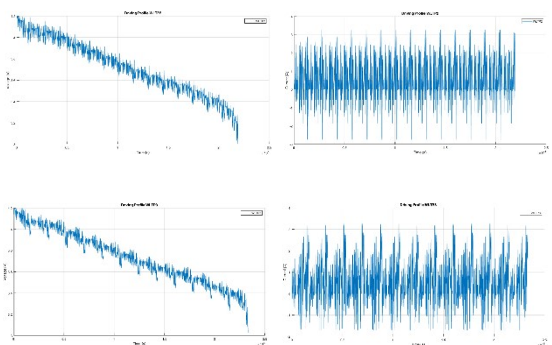


Figure 4 The voltage (left) and current (right) are input data as will be fed into networks. (a) WLTP Class 2 and (b) WLTP Class 3

From Figure 4, the data is used for training and testing the network. WLTP Class 2 data is concatenated 15 cycles, resulting in WLTP Class 2 driving profile data with a total of 21,900 data points. WLTP Class 3 data is concatenated 13 cycles, resulting in WLTP driving profile data. Class 3 has 23,322 data points. Because require the networks to be able to realistically reach the behavior of the battery. In addition, to make the SoC estimation in this research as very close as possible to the actual implementation with the data-driven method.

However, each class of the WLTP driving profile dataset is divided into three proportions. The training set: is used for training the network; The validation set: is used for post-training testing to see how well the model performs after each adjustment to find the model that works best; The test set: is used for testing after getting the best model, and how well the model will perform with data never seen before; In this experiment, the data is divided into 70% training, 15% validation, and 15% testing, which is the default MATLAB's data allocation, as shown in the footers of Tables 3 and 5.

3.0 RESULTS AND DISCUSSION

As discussed before, the data used to train and test the FNN and LSTM networks were the WLTP class 2 and class 3 driving profiles. these were recorded at a frequency of 1 Hz, and the number of data points was about 20,000. However, the testing was divided into two subsections, SoC estimation with FNN and LSTM networks, respectively. Both network types are experimentally performed according to the experimental conditions as shown in Table 2. Hence, four experimental cases were tested in this study, the author will not set any initial parameters to train networks. This trains the entire network for 100 epochs, 100 iterations, or 1 iteration per epoch, 10 Minibatch Size, and assigns the same 10 hidden units for equality in comparison. Then, train and test the network according to the conditions.

In this paper, the FNN and RNN-LSTM networks are evaluated using error functions for the SoC estimation performance of the network. These include Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) in each test case with WLTP driving profiles class 2 and class 3.

3.1 FNN for SoC Estimation

This subsection will discuss the Estimated SoC performance of the FNN after the network was trained with the data for each test case. The network structure of the FNN has one input layer that consists of voltages and currents, 2 hidden layers, each hidden layer having 10 computation nodes, and using hyperbolic tangent as an activation function. The output layer is regression and has a clippedRelu setting so that the answer cannot be greater than 1 (100% SoC) is an activation function, the architecture, and data proportion of FNN in this experiment are indicated in Table 3.

Table 3 Hyperparameters for the FNN model

Layers	Names	Number of Neurons
1	sequenceInputLayer	2
2	fullyConnectedLayer (1)	10
3	tanhLayer	1
4	fullyConnectedLayer (2)	10
5	tanhLayer	1
6	fullyConnectedLayer (3)	1
7	clippedReluLayer	1
8	regressionLayer	1
Train data	**Validation data**	**Test data**
70%	15%	15%

FNN can estimate SoC quickly and fairly accurately. Since the structure of the FNN is simple and there is no connection between the information in each step, FNN can be completed quickly in less than a minute. The FNN training progress plot processed in MATLAB is depicted in Figure 5.

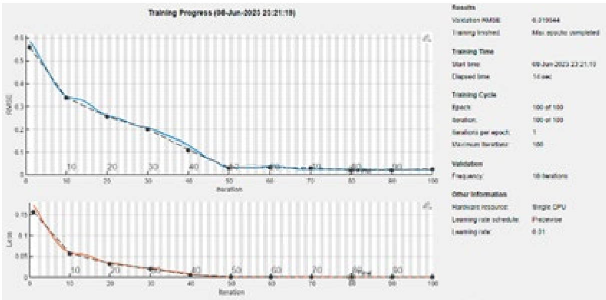


Figure 5 In MATLAB, the FNN training achieved the best RMSE, minimal Loss, and quick training time.

In the initial test case, FNNs were trained and tested using the WLTP class 2 dataset. Surprisingly, with just 14 seconds of training time, the FNN yielded impressive estimation results, as depicted in Figure 6 (a). The RMSE of the network stood at a mere 2.134%, indicating the network's efficiency in SoC estimation despite the relatively short training duration.

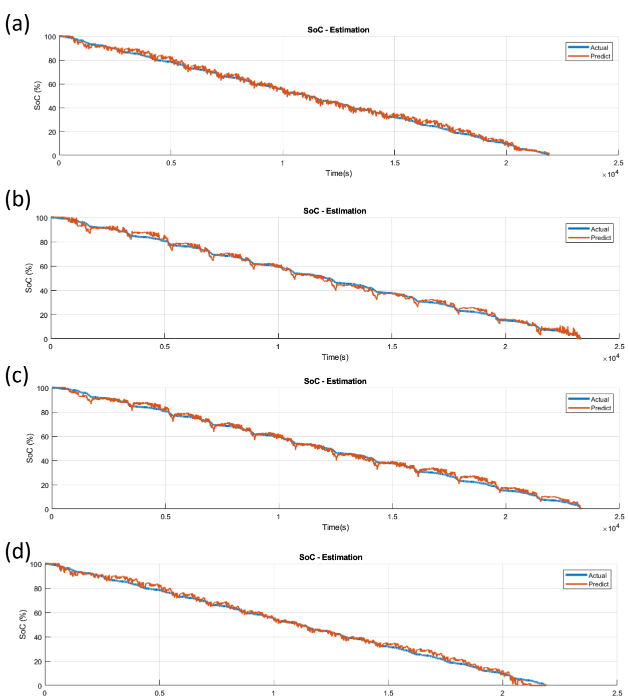


Figure 6 SoC of the battery from FNN estimation performed on (a) test case 1, (b) test case 2, (c) test case 3, and (d) test case 4

Later, used the WLTP class 3 driving profile dataset in the second test case to train and test the FNN for SoC estimation. In this experiment, the network took the same training time as in case 1, 14 seconds, but the performance gain was better. The RMSE is equal to 1.964% which is less than in the first case. Nevertheless, it is worth considering that in this case, the SoC value estimated by the FNN is less accurate compared to the first case, as depicted in Figure 6 (b). It can be noted that the graph in the first case is closer to the true value than in this case, which may be a result of the non-linearity of the WLTP class 3 dataset.

The third case, it's different from the previous two cases. The data used to train and test the network will not be the same, because the data used for training is WLTP class 2 and tested by class 3. FNN takes only 11 seconds to train, this is three seconds faster than the first two cases, and the RMSE of 2.128%, which is considered acceptable performance. It can be seen from Figure 6 (c) that the SoC estimation of FNN, in this case, is satisfactory. So, there will be an error in some phases of the discharge, due to the battery nonlinearity recorded in the WLTP class 3 dataset.

In the final case, use WLTP class 3 to train the network, and WLTP class 2 is used to test the SoC estimation of the FNN. In this test, the network takes the longest training time of 16 seconds and the estimation results obtained in this case are good with an RMSE of 1.997%. The SoC is estimated by FNN, in this case, considering the best SoC estimation result for FNN in this paper. The graphs of the actual SoC and the estimated SoC are quite close, as shown in Figure 6(d), and there is relatively little volatility in the line compared to other test cases.

As a result of the test, FNN was used to estimate the SoC values from all four test cases. It was found that the network constructed to estimate the SoC in this study can work quickly and make estimates quite accurately. However, there will be a slight oscillation due to the nonlinear of the battery, but it is considered that the FNN also provides good estimation results. The performance of the SoC estimation test in different test cases can be seen in Table 4, it appears the error used to evaluate the network's performance is relatively low. So, it can be concluded that FNN created in this study rapidly and accurately works to estimate the SoC of WLTP driving profiles that include class 2 and class 3.

Table 4 SoC estimation performance of FNN trained on each test case

Case	MAE (%)	MSE (%)	RMSE (%)
1	1.689	0.014	2.134
2	1.546	0.003	1.964
3	1.714	0.008	2.128
4	1.596	0.004	1.997

These errors were computed by MATLAB

3.2 SoC Estimation with LSTM

In this subsection, the efficiency of SoC estimation using LSTM is tested by training the network with datasets from 4 test cases. LSTM consists of 4 layers, one sequenceInputLayer with 2 nodes (voltages and currents), a lstmLayer with 10 hidden units in the layer, a connector of nodes is the fullyConnectedLayer, and the regressionLayer is an output of the network, as indicated in Table 5.

Table 5 The structure of LSTM

Layers	Names	Number of Neurons
1	sequenceInputLayer	2
2	lstmLayer	10
3	fullyConnectedLayer	1
4	regressionLayer	1
Train data	**Validation data**	**Test data**
70%	15%	15%

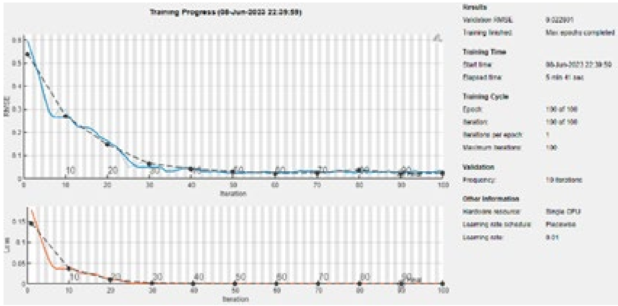


Figure 7 The training progress of LSTM in the second test case showed the best performance.

The LSTM can be accurate estimations but it takes longer to train the network than FNN. Because the LSTM structure also takes the output in the previous period to be the input in the current period as well, to increase the network's efficiency in working with related, sequence, or time-series datasets. Thus, making it more complex and computational. Therefore, it takes more time to train the network, but it can get converged in less than 10 minutes, as presented in Figure 7.

Firstly, the LSTM was trained and tested with the WLTP class 2 dataset. Networks took almost 6 minutes to train the network, which is rather fast, and LSTM can show excellent estimation efficiency. The efficiency of the LSTM can be explained in Figure 8 (a), the actual and estimated values of the LSTM are most closely. Although, there may be a slight oscillation in some parts. However, in this experiment case, the SoC estimation efficiency of the network can be evaluated from the LSTM having an RMSE of 2.356%.

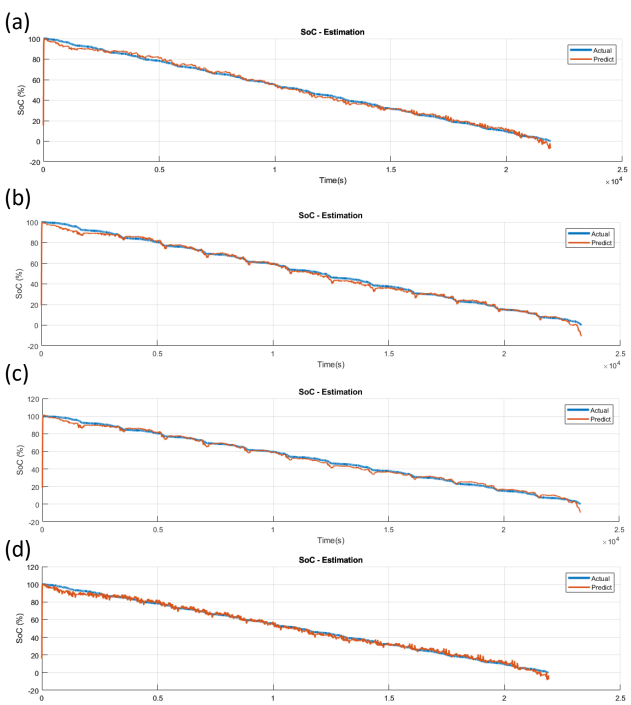


Figure 8 The performance of LSTM for SoC estimation based on (a) test case 1, (b) test case 2, (c) test case 3, and (d) test case 4

In the second case, the LSTM transitioned from training and testing with WLTP class 2 to class 3. The network training time remained similar to the previous case. The network showed good efficiency with an RMSE value of 2.281% which was lower than in the first case. But, when notice the results shown in Figure 8 (b), there was a longer oscillation of the responses than in the first case. This may be due to the nonlinearity of the dataset. Nevertheless, considering at overall, the LSTM still offers acceptable estimation performance, it's quite accurate in many moments.

In the third test case, the experiment is different from the previous two cases. The network is trained on the class 2 dataset and tested with class 3. In this case, the LSTM still exhibits good performance. However, the datasets that the network is trained and tested are not the same. Figure 8 (c) shows the results of SoC estimation obtained by LSTM in the third test case, which has relatively great accuracy. Sometimes, there may be an oscillation in the response at certain times, but most rarely. Therefore, this good performance of the LSTM in this test case can be confirmed by the RMSE, which is equal to 2.339% and is also the smallest in comparison to another test case of the LSTM in the experiment.

Lastly, in the final test case, there was a switch in the training and testing datasets. The LSTM was trained with WLTP class 3 and tested with class 2 to estimate the SoC. The network training time is closely aligned with the training durations in the other test cases. The LSTM still shows good performance with an RMSE value of 2.502%. Then, when considering Figure 8 (d), it can be seen that the estimated SoC value of the LSTM is very close to the true value with a discrepancy in the response only at the start and the end of testing.

Table 6 SoC estimation performance of LSTM trained on each test case

Case	MAE (%)	MSE (%)	RMSE (%)
1	1.739	1.317e-6	2.356
2	1.604	0.006	2.281
3	1.766	0.002	2.339
4	1.762	0.002	2.502

These errors were computed by MATLAB

Through the testing of SoC estimation using LSTM in all the test cases, it was observed that the LSTM model developed in this paper demonstrated accurate and efficient estimation of battery SoC. The model successfully handled the data collected from the WLTP class 2 and class 3 driving profiles. The table showing the efficiency of LSTM in each case is shown in Table 6. It also takes slightly more time to train the network than FNN due to the structure and computational method of the network. Moreover, LSTM has less response oscillation when compared to FNN. But there are quite a lot of errors at the start. Since the experiment in this research did not set the initial parameter of the LSTM, it requires randomizing the initial parameters of the network. Then it adjusts itself to estimate to reach the target of the output data. This is a feature of the RNN with an LSTM cell, it can memorize and modify all the data or parameters of the network by itself. which is considered the highlight of LSTM.

However, when comparing the experimental results obtained from FNN and LSTM in this paper with other studies using different driving profile datasets at 25˚C. It was found that the artificial neural network of this research still has inferior performance. This can be observed from the average

RMSE value because WLTP is a new standard for testing electric cars. That still has relatively little usage profile information (3 classes) compared to the old standard like NEDC can be divided into many sub-profiles such as UDDS, US06, LA92, and HWFET, making the network of research [12, 13] receive more training. This results in a more accurate estimation of the battery's charge status. Including the differences in the structure within the network. But this research's neural network can converge faster. It only takes a few minutes compared to the amount of data used for the network, as shown in Table 7.

Table 7 Comparison of SoC estimation performance

Networks	Driving Profiles	Avg. RMSE
FNN	WLTP Class 2, WLTP Class 3	2.056
FNN [12]	UDDS, HWFET, LA92	1.4
LSTM	WLTP Class 2, WLTP Class 3	2.367
LSTM [13]	HWFET, UDDS, LA92, US06	1.110

4.0 CONCLUSION

An analysis of the results in this research, the performance of both networks is mainly taken into account in SoC estimation, divided into performance function or error function analysis, and analysis of the FNN and LSTM estimation result curves from the four test cases as shown in Table 2.

Considering the estimation results obtained from both networks using the error function. FNN had mean RMSE, MSE, and MAE values across the four cases of 2.056%, 0.007%, and 1.636%, respectively. Meanwhile, LSTM was 2.369%, 0.002%, and 1.718%, respectively. Furthermore, FNN took faster training and converge than LSTM, which may be due to the complexity of the network structure. The results indicate that LSTM involves more computations, resulting in longer processing times. Additionally, upon analyzing the result curves presented in Figures 6 and 8, it is evident that LSTM outperforms FNN. The LSTM demonstrates better accuracy as its response closely aligns with the actual values and exhibits less oscillation compared to FNN. However, FNN will have intermittent oscillations throughout the run, whereas LSTM will only indicate an early error. Due to the initial parameters randomization of the network, then it adapts to the dataset. Hence, the response during the process exhibits minimal deviation from the desired values. Since, the large number of errors at the cycle start of the LSTM. As a result, the RMSE, MSE, and MAE values obtained from the LSTM, as presented in MATLAB, are higher compared to those of the FNN.

Another reason is that makes FNN and LSTM cannot fully show SoC estimation performance. Since, there may be oscillation in the initial and end of responses, or they may occur throughout the process in the case of FNNs, possibly as a result of the nonlinearity of the battery. This also makes the recorded data set non-linear, and another important thing is that the datasets used to train the networks are too few (only has WLTP class 2 and class 3). But, because of the properties of the LSTM, it can adapt and estimate more closely, whereas FNN cannot.

In conclusion, the analysis results indicate that FNNs have a simpler network structure, enabling quick and efficient training for straightforward problems with lower computational requirements. On the other hand, LSTM excels in handling sequential data and has the ability to learn and enhance network parameters autonomously. This makes the LSTM suitable for SoC estimation, due to SoC estimation requires knowing the previous SoC remaining within the battery to increase the accuracy and reliability of the battery's usage.

Future works have a plan to explore different temperature conditions, including those above 25˚C, to enhance the realism of the networks. This is important as electric vehicle batteries can experience temperatures higher than 25˚C during operation. Additionally, intend to implement filters to eliminate noise from measurement data, ensuring more accurate and reliable results.

Acknowledgment

The authors gratefully thank K. Rupanwong for searching and collecting WLTP driving profiles and recording the datasets of the battery discharge based on WLTP class 2 and class 3 using a thermal temperature chamber at the Department of Electrical Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper

References

[1] H. Nazaripouya, B. Wang, and D. Black. 2019. Electric Vehicles and Climate Change: Additional Contribution and Improved Economic Justification. *IEEE Electrification Magazine*. 7(2): 33–39. DOI: https://doi.org/10.1109/MELE.2019.2908792.

[2] E. Chemali, M. Preindl, P. Malysz, and A. Emadi. 2016. Electrochemical and electrostatic energy storage and management systems for electric drive vehicles: State-of-the-art review and future trends. *IEEE Journal of Emerging and Selected Topics in Power Electronics*. 4(3): 1117-1134. DOI: https://doi.org/10.1109/JESTPE.2016.2566583.

[3] R. Ahmed, M. El Sayed, I. Arasaratnam, J. Tjong, and S. Habibi. 2014. Reduced-Order Electrochemical Model Parameters Identification and State of Charge Estimation for Healthy and Aged Li-Ion Batteries - Part II: Aged Battery Model and State of Charge Estimation. *IEEE Journal of Emerging and Selected Topics in Power Electronics*. 2(3): 678–690. DOI: https://doi.org/10.1109/JESTPE.2014.2331062

[4] Z. Deng, L. Yang, Y. Cai, H. Deng, and L. Sun. 2016. Online available capacity prediction and state of charge estimation based on advanced data-driven algorithms for lithium iron phosphate battery. *Energy*. 112: 469–480. DOI: https://doi.org/10.1016/j.energy.2016.06.130.

[5] Bebis G., and M. Georgiopoulos. 1994. Feed-forward neural networks. *IEEE Potentials*. 13(4): 27–31. DOI: https://doi.org/10.1109/45.329294.

[6] Dheeraj Kumar Dhaked, Sharad Dadhich, and Dinesh Birla. 2023. Power output forecasting of solar photovoltaic plant using LSTM. *Green Energy and Intelligent Transportation*. 2(5): 100113. DOI: https://doi.org/10.1016/j.geits.2023.100113.

[7] Manjot Kaur and Aakash Mohta. 2019. A Review of Deep Learning with Recurrent Neural Network. *International Conference on Smart Systems and Inventive Technology 2019 (ICSSIT).* 460-465. DOI: https://doi.org/10.1109/ICSSIT46314.2019.8987837.

[8] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*. 9(8): 1735-1780.DOI: https://doi.org/10.1162/neco.1997.9.8.1735.

[9] O. Kyung Su, K. Dong Myung. 2014. Product Specification Rechargeable Lithium Ion Battery Model: INR18650 MJ1. *LG CHEM. SEOUL 150-721*. KOREA.

[10] Hyeonjik Lee and Kihyung Lee. 2020. Comparative Evaluation of the Effect of Vehicle Parameters on Fuel Consumption under NEDC and WLTP. *Energies 2020.* 13(16): 4245. DOI: https://doi.org/10.3390/en13164245.

[11] S. Tsiakmakis, G. Fontaras, J. Pavlovic, and C. Cubito. 2017. From NEDC to WLTP: effect on the type-approval CO2 emissions of light-duty vehicles. *JRC Science for Policy Report on European Commission*. EUR 28724 EN.

[12] Romulo N. Vieira, Phillip Kollmeyer and Mina Naguib. 2023. Feedforward and NARX Neural Network Battery State of Charge Estimation with Robustness to Current Sensor Error. *2023 IEEE Transportation Electrification Conference & Expo (ITEC).* DOI: https://doi.org/10.1109/ITEC55900.2023.10187084.

[13] Ephrem Chemali, Phillip J Kollmeyer and Ryan Ahmed. 2017. Long Short-Term Memory-Networks for Accurate State of Charge Estimation of Li-ion Batteries. *Ieee Transactions On Industrial Electronics.* 65: 6730.6739. DOI: https://doi.org/10.1109/TIE.2017.2787586.