Article

Deep Learning with Spatial Attention-Based CONV-LSTM for SOC Estimation of Lithium-Ion Batteries

Huixin Tian [1,2,*] and Jianhua Chen [1,2]

1 School of Control Science and Engineering, Tiangong University, Tianjin 300387, China
2 Tianjin Key Laboratory of Intelligent Control of Electrical Equipment, Tiangong University, Tianjin 300387, China
* Correspondence: tianhuixin@tiangong.edu.cn

Abstract: Accurate estimation of the state of charge (SOC) is an indispensable part of a vehicle management system. The accurate estimation of SOC can ensure the safe and reliable operation of the vehicle management system. With the development of intelligent transportation systems (ITS), vehicles can not only obtain the dynamic changes inside the battery through sensors, but also obtain the traffic information around the vehicle through vehicle–road collaboration. In addition, the development of onboard graphic processing units (GPUs) and Internet of Vehicles (IOV) technology make the computing power of vehicles no longer limited by hardware, which makes neural networks applied to the intelligent control of vehicles. Aiming at the problem that the traditional network cannot effectively obtain the complex spatial information of sample attributes, we developed an attention-based CONV-LSTM module for SOC prediction based on a convolutional neural network (CNN) and a long short-term memory (LSTM) network. Different from the traditional LSTM network, the algorithm not only considers the temporal correlation of the data stream, but also captures the spatial correlation information of the input data by convolution. It then uses the different weights, automatically assigned by the attention mechanism, to correctly distinguish the importance of different input data streams. In order to verify the validity of the model, this paper selects the degradation data set of the aeroengine as the verification data set. Experiments show that the proposed model has achieved good results. Finally, the proposed model is applied to the actual vehicle running data, and the effectiveness of the proposed model is verified by comparing it with the Multi-Layer Perceptron (MLP), LSTM, and CNN-LSTM models.

Keywords: IOV; lithium-ion battery; SOC; deep learning

1. Introduction

Due to ongoing issues with oil supply and increasing environmental pollution, electric vehicles (EVs) have gradually supplanted regular cars as the primary mode of transportation [1]. Because of its high energy density, low self-discharge rate, and numerous cycles, lithium-ion batteries are frequently utilized in electric cars [2]. However, when lithium-ion batteries are used incorrectly or are not changed on time, the pleasurability of driving and even the safety of passengers suffer. As a result, an efficient and safe battery management system (BMS) is required to monitor the battery's status and characteristics. SOC, as one of the essential BMS assessment indexes [3], may be used to monitor the remaining capacity of the battery and as a characterization to assure the vehicle's steady operation. SOC is commonly described as the ratio of the battery's current remaining capacity to its maximum capacity [4]. SOC cannot be directly measured but can only be calculated by algorithms using observable variables (such as current, voltage, and temperature). Accurate prediction of SOC remains difficult due to the complex dynamic changes inside the battery and the vehicle's increasingly complicated exterior environment, which manifest in factors such as the battery self-discharge rate, power regeneration, and driving conditions [5]. In the literature, the main methods of SOC estimation can be divided

into three main categories [6], which are traditional methods, model-based methods, and data-driven methods.

The open-circuit voltage (OCV) method [7] and the ampere-hour integral method [8] are two traditional methods. The ampere-hour integral method, also known as the Coulomb counting method, estimates the battery's state of charge (SOC) by integrating the charging/discharging current at the terminals of the battery during operation, and does not require a battery model. Kong Soon Ng [9] proposed an intelligent estimation method for lithium-ion battery SOC and SOH based on Coulomb counting. Although the ampere-hour integral method is simple to implement and produces direct results, it requires high sensor accuracy and that the working conditions of the battery do not change significantly. However, in the actual operation process, the working conditions of the electric vehicle are more complicated, making it impossible for the current to reach a continuous and stable state. In addition, the initial SOC of the battery is also directly linked to the results, so the means to obtain the initial value of SOC is also a problem that has always existed in this method. The open-circuit voltage (OCV) method [7] is a direct calculation based on the relationship between the battery's SOC and its OCV. Although this process is very simple, to correctly measure the OCV of the battery, it is necessary to stand the battery for some time to make the internal electrolyte evenly distributed, which makes the open circuit voltage method unsuitable for online applications.

Model-based estimation methods are also often used in battery SOC prediction. Common methods include EM (electrochemical model), EIM (electrochemical impedance model), and ECM (equivalent circuit model). These methods are based on the reaction principle inside the battery and use mathematical formulas and electrical components to simulate the battery. These methods require the modeler to have a strong knowledge of the battery, and when the accuracy is improved, the parameters of the battery model also increase. In addition, in order to improve the nonlinear and dynamic performance of the model, researchers often combine the nonlinear observer with the battery model. The commonly used methods are the Kalman filtering method and its improved algorithm [10]. In this method, the researchers regard the battery as a whole, without considering the internal reaction principle of the battery. According to the error between the terminal voltage of the model and the measured voltage, the difference is fed back to the predicted value of the SoC through a gain matrix. After multiple recursions, the voltage output of the model is changed to minimize the voltage error. However, this technology is very dependent on the accuracy of the circuit model, which will directly lead to the increased complexity of the algorithm. In the paper [11], a battery SOC prediction method based on the autocovariance least-squares technique and an unscented Kalman filter is proposed. The model is constructed using the state space model of the RC equivalent circuit, and then the SOC prediction model is modified by using the improved unscented Kalman filter of autocovariance least-squares. Zou et al. [12] used partial differential equations to represent the dynamics of batteries under various operating conditions. The Kalman filter is an iterative algorithm. Its final accuracy depends on the accuracy of the battery model. At the same time, the equivalent circuit and observer need to be re-established for different batteries. These problems are the shortcomings of the model-based method.

In recent years, with the development of machine learning, the data-driven SOC prediction method has gradually attracted the attention of researchers. The data-based estimation method refers to the method of automatically learning some relationship between the battery and SOC by measuring the characteristics of the battery related to SOC prediction, such as current, voltage, temperature and internal resistance, etc. The data-driven method has excellent nonlinear modeling capabilities, and data-based modeling does not require the establishment of complex circuit models, which greatly reduces the complexity of the model. The commonly used data-driven methods for SoC estimation include SVM (support vector machine)-, ELM (extreme learning machine)-, RF (random forest)-, and neural network (ANN)-based methods. In [13], the least-squares method is used to reduce the dimension of high-dimensional data for input data, and then the parameters of SVM are

optimized by differential evolution and weighted regularization. Finally, the performance of the prediction model is evaluated by cross-validation, and the SoC prediction problem is transformed into a nonlinear regression problem. Li et al. [14] created a SOC prediction approach based on a random forest (RF) algorithm and conducted experiments with varying discharge currents. LIU [15] used principal component analysis (PSO) and particle swarm optimization to improve the accuracy and robustness of BPNN (back propagation neural network) and then applied the proposed algorithm to the SOC prediction of the battery. The final model achieved good results under different working conditions. Although the commonly used machine learning methods are widely used, they also have certain limitations. For example, when there are many variables in the training data and the relationship between the data is more complex, the model cannot achieve the desired effect.

With the continuous development of cloud computing and computer technology, more and more methods based on artificial neural networks have attracted people's attention. Among them, RNN (recurrent neural network) and its variants are the most commonly used prediction methods. Compared with traditional neural networks, RNN is more suitable for battery SOC prediction. Because the SOC of the current battery has a strong correlation with the previous SOC, the unique 'gate' structure of RNN can save the historical data of the battery for the next SOC prediction. Therefore, RNN can play a better role in battery SOC prediction. Chao et al. [16] applied RNN to the SOC and SOH prediction of lithium-ion batteries and validated the comparison between RNN and other methods across several data sets. The article [17] used GRU (gate recurrent unit)-RNN for battery SOC prediction, obtaining good results using only the measured current, voltage, and temperature. Yang et al. [18] successfully employed LSTM to SOC prediction to better determine the time series characteristics of features.

Although the traditional RNN and its improved network can make use of the time connections between input data, they cannot obtain the correlation between features well when facing high-dimensional input data. At the same time, the standard model does not distinguish input features and assumes that all inputs contribute equally to the prediction results. As a result, in order to address the aforementioned two issues, this study develops a CONV-LSTM time series prediction approach with an attention mechanism. The suggested model can not only obtain the correlation between features automatically, but it can also automatically assign different attention coefficients based on the relevance of input features to output features. This paper's innovations are as follows:

(1) A new CONV-LSTM deep prediction network structure is suggested and applied to multitime series data prediction tasks based on convolution and LSTM networks.
(2) For more accurate prediction, the spatial attention mechanism is introduced to the model, which may choose the relevant input features in each time step.
(3) The suggested model is evaluated using two actual multivariate time series data sets to demonstrate its usefulness.

The remainder of the article is organized as follows. The relevant theoretical information is introduced in Section 2. Section 3 goes over the proposed method in depth. Section 4 is dedicated to experimental validation and analysis. Finally, Section 5 summarizes the article's conclusion.

2. Background

2.1. LSTM Neural Network

Long short-term memory recurrent neural network (LSTM) is a popular time series modeling approach. Hochreiter [19] proposed it in 1997 to answer the problem of recurrent neural network (RNN) gradient disappearance and explosion. The LSTM network is completed by adding a 'gating mechanism' to the present input and historical data memory filter, as shown in Figure 1.
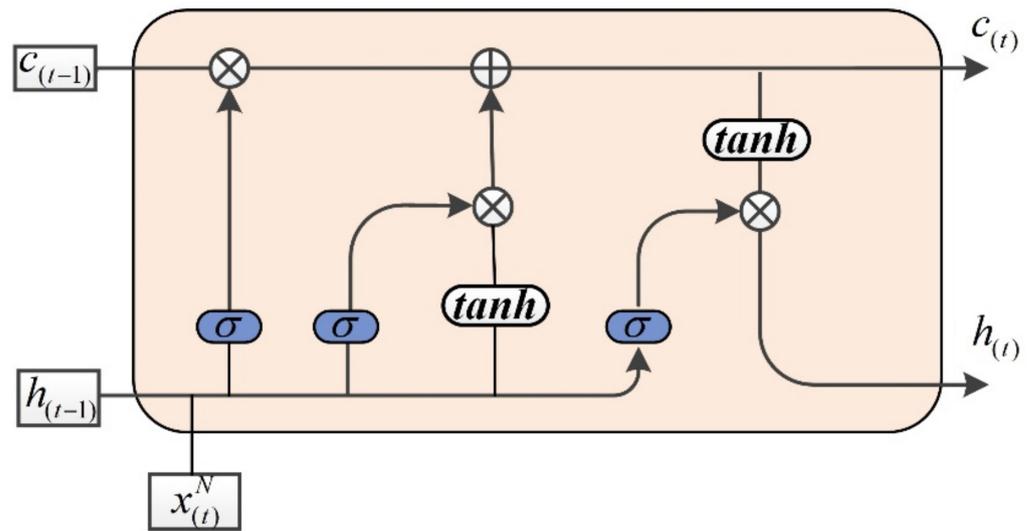
Figure 1. LSTM schematic.

A single LSTM unit operation formula can be described as follows [20]:

$$
\begin{aligned}
f_t &= \sigma(W_f[x_{(t)}, h_{t-1}] + b_f) \\
i_t &= \sigma(W_i[x_{(t)}, h_{t-1}] + b_i) \\
\widetilde{c}_t &= \sigma(W_c[x_{(t)}, h_{t-1}] + b_c) \\
c_t &= f_t c_{t-1} + i_t \widetilde{c}_t \\
o_t &= \sigma(W_o[x_{(t)}, h_{t-1}] + b_o) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
\tag{1}
$$

where $x_{(t)}$ is the input data currently transmitted to the LSTM network, h_{t-1} is the hidden state of the LSTM at the previous moment, [•] is the stitching operation of the matrix, σ is the activation function, *tanh* is the arctangent activation function, and represents the *Hadamard* multiplication. In addition, W_f, W_i, W_c and W_o are parameters that LSTM operations need to learn, where f_t, i_t, $\widetilde{c}_t$, c_t and o_t are the 'gate' structures in the LSTM network, respectively.

2.2. Attention Mechanism

The attention mechanism has been widely discussed and studied in recent years. It works similarly to the human visual system. When the network is trained, it can focus more on the input that is most related to the prediction target and suppress the unrelated content. At present, the attention mechanism is widely used in image classification [21], speech recognition [22], and machine vision [23], and people also extend the attention mechanism to data processing. DARNN [24] integrates the attention mechanism with the LSTM network and assigns different attention coefficients to the input results based on their contribution, boosting the LSTM's accuracy and resilience.

The working principle of the attention mechanism is shown in Figure 2. Suppose N input sequences $x^k = (x_1^k, x_2^k, \cdots, x_N^k)^T \in \mathbb{R}^T$ are given, where k is the dimension of the input sequence and N is the length of the data. To select the most favorable $x_i^k, i \in [1, N]$ for the target, the self-attention distribution coefficient is calculated using the input data x^k and the task-related query set vector q.
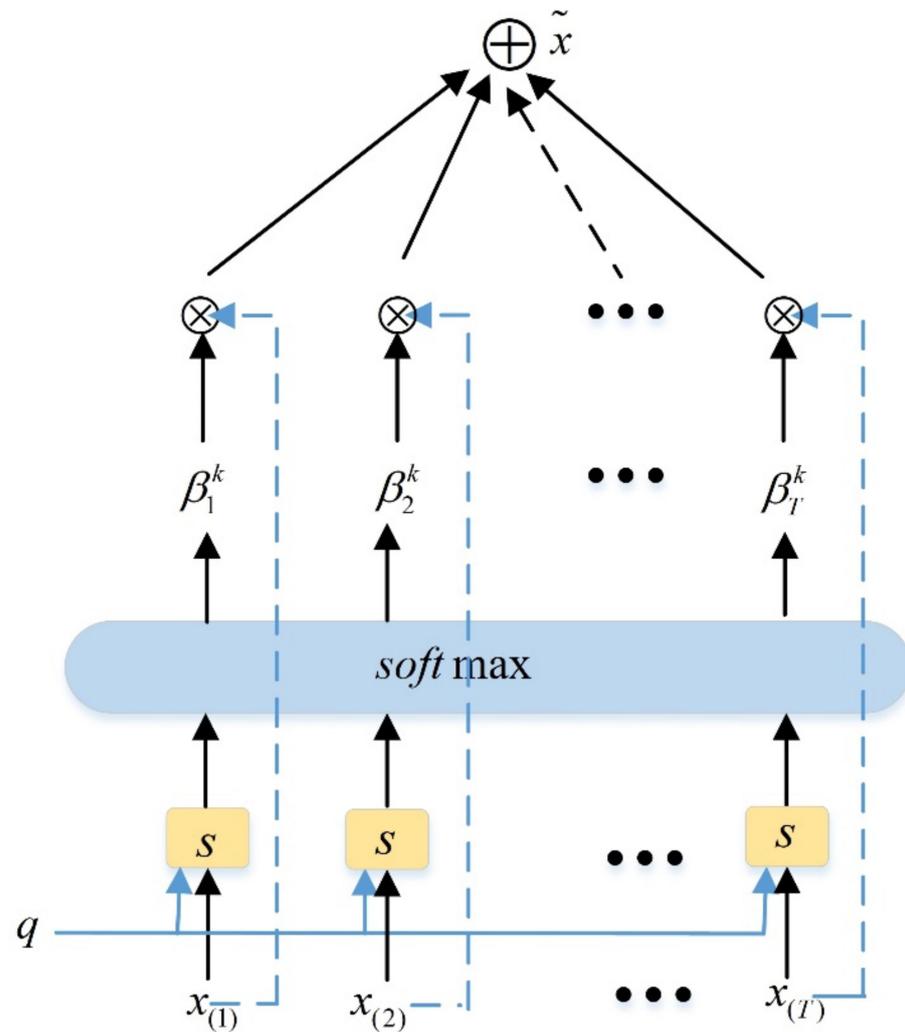
Figure 2. Attention mechanism schematic.

$$\beta_t^k = soft\text{max}(s(x_i^k, q)) = \frac{\exp(s(x_i^k, q))}{\sum_{i=1}^{n} \exp(s(x_i^k, q))} \tag{2}$$

Among them, β_t^k represents the attention coefficient x_i^k, and $s(x_i^k, q)$ is a function of calculating the attention score, which is generally defined as:

$$s = v_e^T \sigma(W_e x_i^k + U_e q + b_e) \tag{3}$$

v_e^T, W_e, U_e and b_e are parameters that can be learned with the overall network. Their detailed meanings are described in Section 3.

3. AT-CONVLSTM

CNN-LSTM has been used as a hybrid model in traffic flow prediction [25], weather prediction [26], and other applications. The model combines CNN's local feature extraction ability with LSTM's temporal feature extraction ability, which improves the model's spatial–temporal extraction capacity to some level. However, it solely analyzes the spatial information between the input characteristics within each time interval and overlooks the transfer of temporal–spatial information. As a result, this research introduces AT-CONVLSTM, a new attention-based spatial–temporal information extraction network. Figure 3 depicts the AT-CONVLSTM schematic.
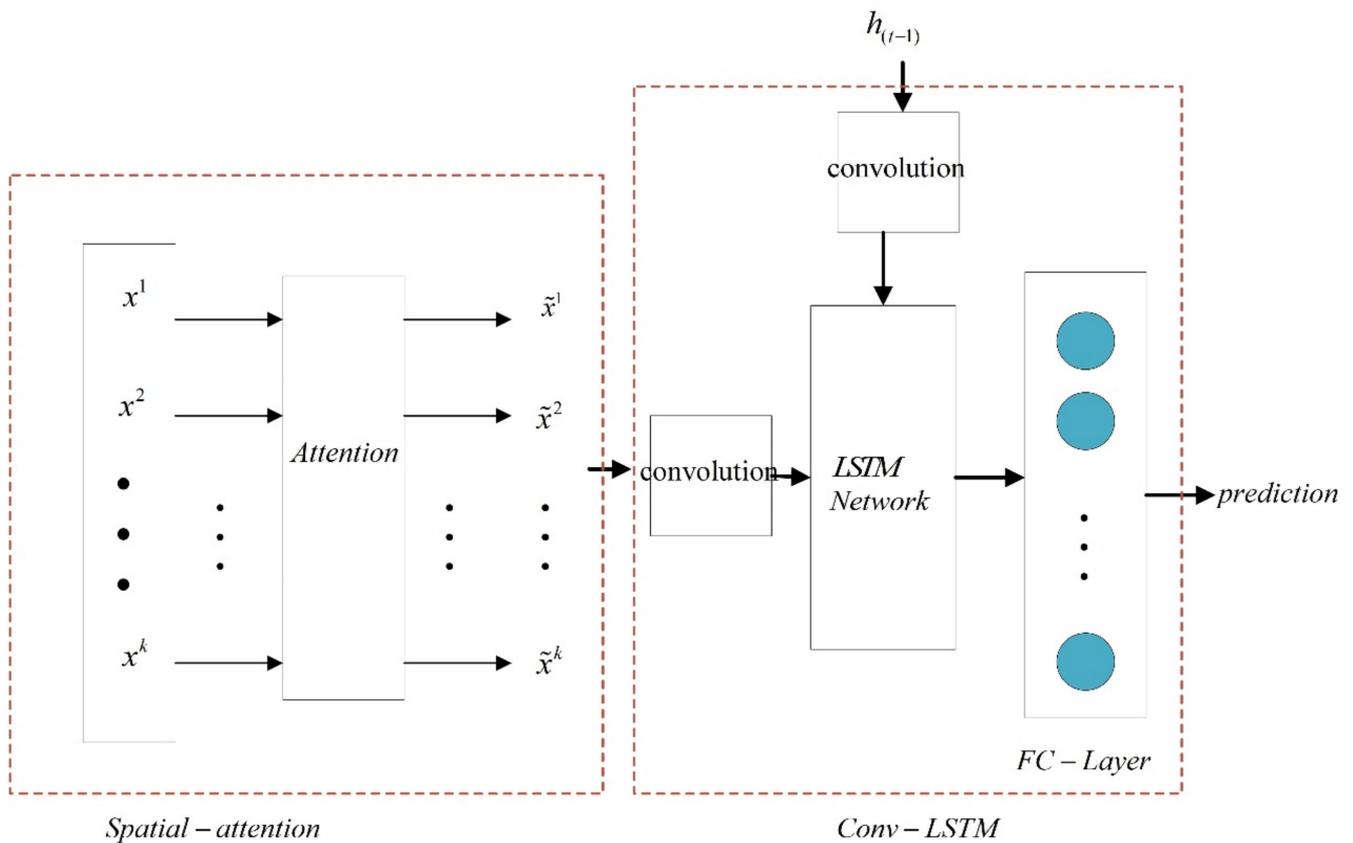
Figure 3. Framework of the proposed method.

The AT-CONVLSTM network consists of a spatial attention module and a Conv-LSTM module. At each time step, the spatial attention module can assign different attention coefficients to the input data features and alter the weights of distinct input features adaptively. The Conv-LSTM module improves the LSTM network's spatio-temporal feature extraction ability for input data in each time step by utilizing the powerful local feature extraction ability of the convolutional network. The following sections provide an overview of the model's many modules.

3.1. Spatial Attention Module

The purpose of this module is to enable the network to adaptively assign different attention coefficients to the input data features of each time step, by which the weights of different input features can be adjusted. Figure 4 is a schematic spatial attention module. It is assumed that the input data flow of each time step model is $X^k = (x_1^k, x_1^k, \cdots, x_T^k)^T \in R^T$, where k represents the dimension of the input data and T represents the window size of the input data. Different color blocks in β_t represent the proportion of different input features at the current moment. The deterministic attention model is used to assess the relationship between the input and the target output produced by the previous layer of LSTM's hidden state. The attention model's formula is as follows:

$$s_t = v_e^T \sigma(W_e h_{t-1} + U_e x^k + b_e) \tag{4}$$

$$\beta_t = softmax(s_t(h_{t-1}, x_t)) = \frac{\exp(s_t)}{\sum_{i=1}^{n} \exp(s_t)} \tag{5}$$
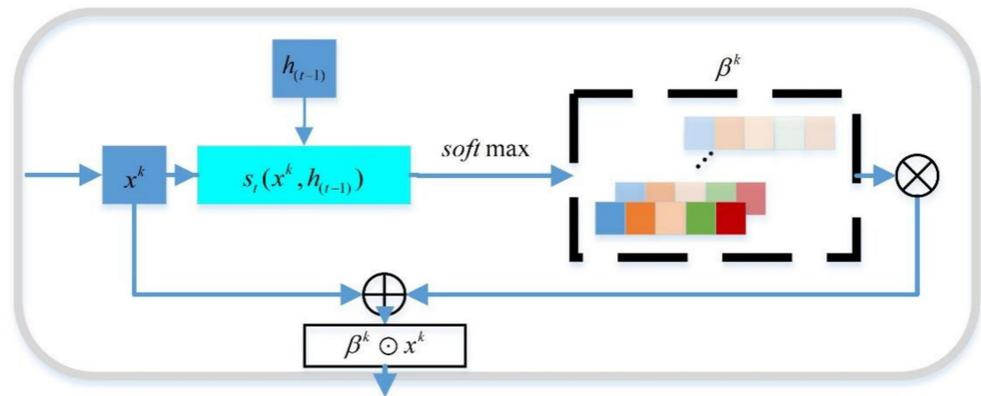
Figure 4. Spatial Attention Module.

In Equation (4), parameters $v_e \in R^T$, $W_e \in R^{T \times 2m}$, and $U_e \in R^{T \times T}$ are parameters that the model needs to learn; σ is a *sigmoid* activation function; s_t is a score function that calculates attention at time t, indicating the importance of each input variable to the predicted output value at time t; h_{t-1} is the hidden state of the LSTM at time $t-1$, and x^k is the input data stream at time t. In Formula (5), β_t is the score function s_t obtained by Softmax function operation, and its purpose is to make the sum of the function be 1. The parameters in the above formula are updated using the backpropagation of the model. The resulting attention coefficient is the spatial attention coefficient, which is multiplied by the input variable x_t to obtain $\widetilde{x}_{(t)}$. The calculation formula is shown in Formula (6):

$$\widetilde{x}_{(t)} = (\alpha_{(t)}^1 x_{(t)}^1, \alpha_{(t)}^2 x_{(t)}^2, \cdots, \alpha_{(t)}^n x_{(t)}^n) \tag{6}$$

The attention module is a feedforward neural network that may complete parameter optimization adaptively with the training of other modules in the model, applying varying degrees of attention to the input data.

3.2. ConvLSTM Modules

The principle of ConvLSTM is shown in Figure 5. The ConvLSTM network replaces the matrix operation in traditional LSTM with convolution operation and uses the feature extraction ability of the convolution network to extract the coupling information of input and hidden state, in order to better improve the spatial feature extraction ability of LSTM.
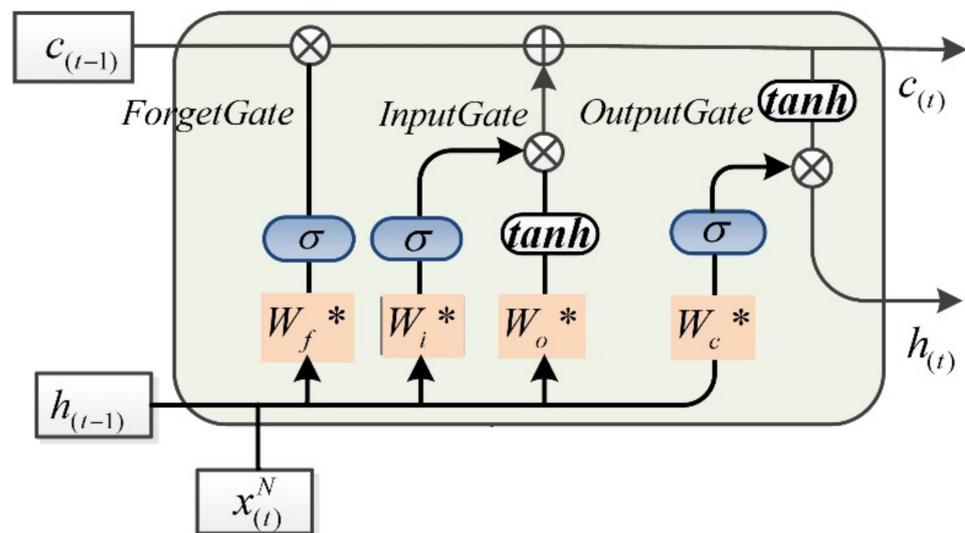

Figure 5. ConvLSTM Module.

The specific formulas of the module are shown in (4) to (9):

$$f_t = \sigma(W_f * [\widetilde{x}_{(t)}, h_{t-1}] + b_f) \tag{7}$$

$$i_t = \sigma(W_i * [\widetilde{x}_{(t)}, h_{t-1}] + b_i) \tag{8}$$

$$\widetilde{c}_t = \sigma(W_c * [\widetilde{x}_{(t)}, h_{t-1}] + b_c) \tag{9}$$

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

$$o_t = \sigma(W_o * [\widetilde{x}_{(t)}, h_{t-1}] + b_o) \tag{11}$$

$$h_t = o_t \bullet \tanh(c_t) \tag{12}$$

In the above formula, $\widetilde{x}_{(t)}$ represents the input data weighted by the spatial attention coefficient, $*$ represents the convolution operation, and the remaining parameters have the same meaning as Formula (1).

Formula (7) represents the calculation method of the "forgetting gate" in the proposed model. Firstly, h_{t-1} and $\widetilde{x}_{(t)}$ are concatenated for convolution operation, and then f_t are obtained by the activation function, which determines what information from the previous moment has been forgotten.

Formula (8) represents the "input gate" of the model. Firstly, the hidden state h_{t-1} and input $\widetilde{x}_{(t)}$ are used for matrix splicing, and then the convolution operation with the parameter W_i is used to extract the features. The extracted information is passed through the sigmoid network to obtain the output matrix i_t, i_t to determine which information needs to be updated.

In Formula (9), $\widetilde{c}_t$ is the intermediate variable of memory unit update, which is the same as the calculation process of the above "gates". The spliced matrix and the weight matrix W_c are multiplied by the dot product, and then the output value is obtained by the tanh activation function.

Formula (10) represents the memory unit update of the model. The process of updating the old memory unit c_{t-1} to c_t can be regarded as consisting of three parts. The first step is the multiplication of c_{t-1} and f_t, which is undertaken to forget the old memory unit that needs to be discarded. Then i_t and $\widetilde{c}_t$ are multiplied to obtain the effective information that needs to be retained at the current time. Finally, the results of the two are added and the updated memory unit state value c_t is obtained.

Formula (11) represents the "output gate". The spliced matrix is extracted by a convolutional network with a parameter of W_o, and then o_t is obtained through the sigmoid network, which determines which part of the information in the cell state can be output. Finally, the output values of o_t and c_t, attained through the activation function tanh are multiplied to obtain the output result h_t of the hidden layer.

4. Experimental Results and Analysis

In this part, we performed two experiments. The RUL prediction experiment of the turbo engine is to illustrate the effectiveness of the proposed AT-CONVLSTM model, which is then applied to the SOC prediction of vehicle battery data [27]. The RUL prediction of the turbofan engine uses the CMAPSS data set published by NASA [28]. The data set is divided into four subdata sets according to the engine failure mechanism and operating conditions. Each training set and test set is taken from a certain point in the life cycle to the end of life. Under the initial conditions of the engine, the degree of mechanical wear is low, which is defined as being in a healthy state. When a fault occurs at a certain point, the performance of the engine will gradually decrease with time, and the current corresponding RUL value is marked at each moment in the data set. The specific information of the data set is shown in Table 1; each data set records the changes of 24 aeroengine variables in each flight cycle, of which 21 are engine sensor measurement variables and 3 are working condition variables. The driving data of the vehicle are derived from the actual driving data of the electric vehicle. The data are all from NEDC (New European driving cycle), with a

maximum speed of 120 km/h and an average speed of 36.1 km/h. The experimental acquisition interval is 0.3 s, and the initial SOC of the vehicle is 70%. The data are composed of 13 variables such as voltage, current, temperature and speed, and record the running state of the vehicle.

Table 1. Information of CMAPSS data set.

	C-MAPSS			
	FD001	**FD002**	**FD003**	**FD004**
Engines in training data	100	160	100	249
Engines in testing data	100	259	100	248
Operation modes	1	6	1	6
Fault modes	1	1	2	2

All experiments are completed under the processor Inter Core i7-6700 model and 32 G RAM Windows 10 operating system. In addition, the tool used was "python 3.6.13", with "PyTorch 1.9.1" as the backend. To better obtain the optimal solution in the training process, this paper adopts the mechanism of 'early termination'. Specifically, 20% is randomly selected from the training data set as the verification set, and the model parameters are updated according to the error of the verification set during the training process. In addition, extensive experiments are conducted to obtain the appropriate parameters, and a grid search strategy is used to determine the appropriate hyperparameters.

4.1. Evaluation Metrics

In order to quantitatively analyze the superiority of the proposed algorithm compared with other algorithms, the article uses root-mean-square error (RMSE) and score function to undertake the performance evaluation of RUL, where the score function is proposed by the 2008 prognosis and health management (PHM) data challenge [27]. This is frequently used as a widely accepted predictor. Similarly, root-mean-square error (RMSE) and mean absolute error (MAE) are utilized as evaluation indicators in vehicle driving data to quantitatively examine the results of the AT-CONVLSTM network and other algorithms. These two error evaluation methods are commonly utilized in regression task evaluation. The RMSE, MAE, and score computation formulas are as follows.

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (m_n - \hat{m}_n)^2} \tag{13}$$

$$MAE = \frac{1}{N} \sum_{n=1}^{N} \left(\left| m_n - \hat{m}_n \right| \right) \tag{14}$$

$$Score = \begin{cases} \sum\limits_{n=1}^{N} \left(e^{-\frac{m_n - \hat{m}_n}{13}} - 1 \right), m_n > \hat{m}_n \\ \sum\limits_{n=1}^{N} \left(e^{\frac{m_n - \hat{m}_n}{10}} - 1 \right), m_n \le \hat{m}_n \end{cases} \tag{15}$$

In the above formula, N is the number of samples, m_n and $\hat{m}_n$ are the predicted value and the actual value of the sample f, respectively. Results with smaller RMSE, MAE, and score values are better than results with larger RMSE and score values.

4.2. Experiment 1: C-MAPSS

Since some sensor data in the C-MAPSS data set do not change with time, it is considered that they do not contain useful information for RUL prediction. Therefore, we selected and used 14 sensor measurements in subsequent experiments. The serial numbers of these 14 sensors are 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20 and 21, respectively [29]. Figure 6 shows one testing sample in FD001 with data of 14 sensors within a time window

of length 60. In order to obtain the best experimental parameters of the comparison test and the proposed method, all experiments have been carried out many times. Finally, the hyperparameters of the related methods are shown in Table 2.
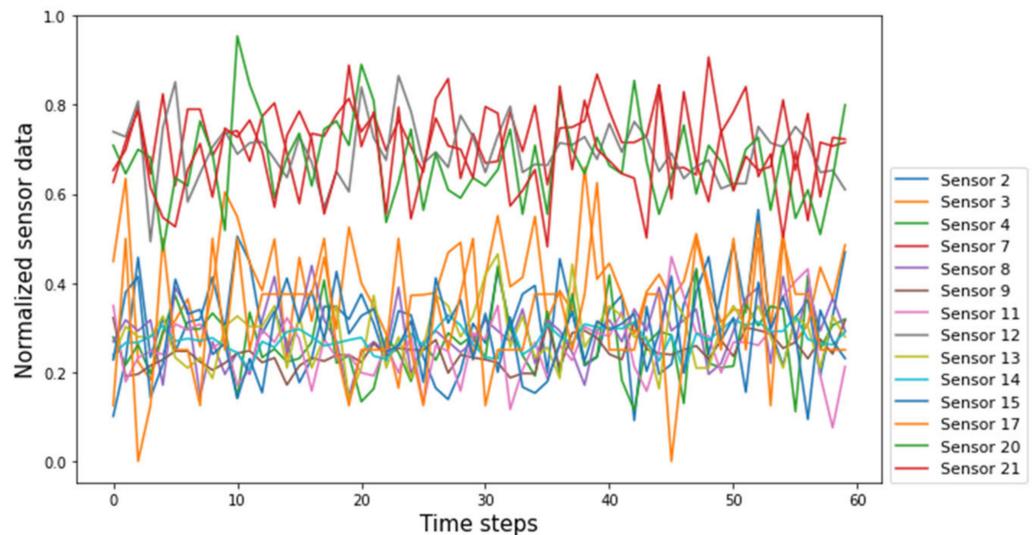

Figure 6. One testing sample in FD001 with data of 14 sensors within a time window of length 60.

Table 2. Parameter settings for each network.

Parameter Value	MLP	LSTM	CNN-LSTM	Conv-LSTM	AT-CONVLSTM
Learning rate	0.01	0.007	0.005	0.007	0.007
Epoch	500	150	150	150	150
Batch size	-	200	200	200	200
Number of hidden layers	2	1	1	1	1
Number of neurons	[60,30,1]	[70,1]	[60,1]	[70,1]	[70,1]
Kernel size	-	-	[3,1]	[5,1]	[5,1]

At the same time, considering the impact of randomness on performance, all experiments in this paper were run 10 times, independently, and then the average taken to determine the final result. The final prediction results of each method are shown in Table 3.

Table 3. Comparison of the average error between the proposed algorithm and other methods.

Testing Data	Metrics	MLP	LSTM	CNN-LSTM	Conv-LSTM	AT-CONVLSTM
FD001	RMSE	20.06	16.67	14.27	14.83	12.95
	Score $(\times 10^5)$	1.6	0.6	0.37	0.39	0.32
FD002	RMSE	27.69	22.53	18.90	18.76	18.51
	Score $(\times 10^5)$	7.1	4.6	2.6	2.4	2.0
FD003	RMSE	19.37	14.63	12.56	11.48	10.98
	Score $(\times 10^5)$	1.6	1.0	0.86	0.45	0.37
FD004	RMSE	22.07	19.93	18.96	17.56	17.06
	Score $(\times 10^5)$	6.3	9.7	7.2	5.7	6.0

Table 3 shows the average error results for the four engine data sets FD001 to FD004 using five different methods. Table 3 shows that MLP has the worse prediction performance across all data sets. Because MLP has a simplistic structure and cannot consider data's temporal relationship, it cannot produce accurate predictions for complex time series data sets. LSTM outperforms MLP in data prediction because it can capture the dynamic character of process data. It can store the nonlinear features extracted by the nonlinear activation function through the loop unit to complete the time extraction function, and then the data

is filtered by the 'gate structure'. When there are multiple characteristics in the data, however, LSTM cannot determine the coupling relationship between them. CNN-LSTM can mine the association between features more effectively than LSTM prediction. Conv-LSTM is a replacement for CNN-LSTM that only addresses the extraction of single-step spatial coupling information of data while neglecting the fact that data coupling information may change over time, hence its effect is superior to CNN-LSTM. The Conv-LSTM module in the proposed AT-CONVLSTM model can not only extract the spatial characteristics of the feature in a single step time, but it can also mine the spatial features that change with time adaptively. In addition, the $Spatial - attention$ module can calculate the attention weight of the input variable by hiding the state variable and the current input, which can better identify the input variables related to the RUL prediction, so the method has higher prediction accuracy. Figure 7 shows the prediction results of MLP, LSTM, CNN-LSTM, and the proposed method on the FD003 data set. It can be seen from Figure 5 that the prediction performance based on MLP, LSTM, and CNN-LSTM is poor. Although their prediction results can follow the actual change trend, the change fluctuates greatly. For predictions based on Conv-LSTM, the prediction curve can better track changes in measurements, but there is still a large deviation between the predicted output and the measured output. For the AT-CONVLSTM model, the predicted results track the actual curve well.

4.3. Experiment 2: Battery SOC Prediction

We use the proposed AT-CONVLSTM model for SOC prediction of lithium-ion batteries. In the experiment, we found that the total voltage and total current of the battery pack have a high correlation with the voltage and current of the single battery by analyzing the correlation coefficient between the sensors. Therefore, we screened the variables with high correlation. The final input variables are the current, voltage, temperature, average current, average voltage, and vehicle speed of the battery for a specified time. The output variable is the SOC of the battery. After many experiments, the parameters of MLP, LSTM, CNN-LSTM, Conv-LSTM, and AT-CONVLSTM used in the article are shown in Table 4.

Table 4. Parameter settings for each network.

Parameter Value	MLP	LSTM	CNN-LSTM	Conv-LSTM	AT-CONVLSTM
Learning rate	0.01	0.007	0.005	0.005	0.005
epoch	500	150	150	150	150
Batch size	-	200	200	200	200
Number of hidden layers	2	1	1	1	1
Number of neurons	[15,40,1]	[35,1]	[35,1]	[35,1]	[35,1]
Kernel size	-	-	[3,1]	[5,1]	[5,1]

Considering the influence of randomness on performance, the experiment was also run independently 10 times, and the final result was obtained by averaging. The final prediction results of each method are shown in Table 5.

Table 5. Comparison of the average error between the proposed algorithm and other methods.

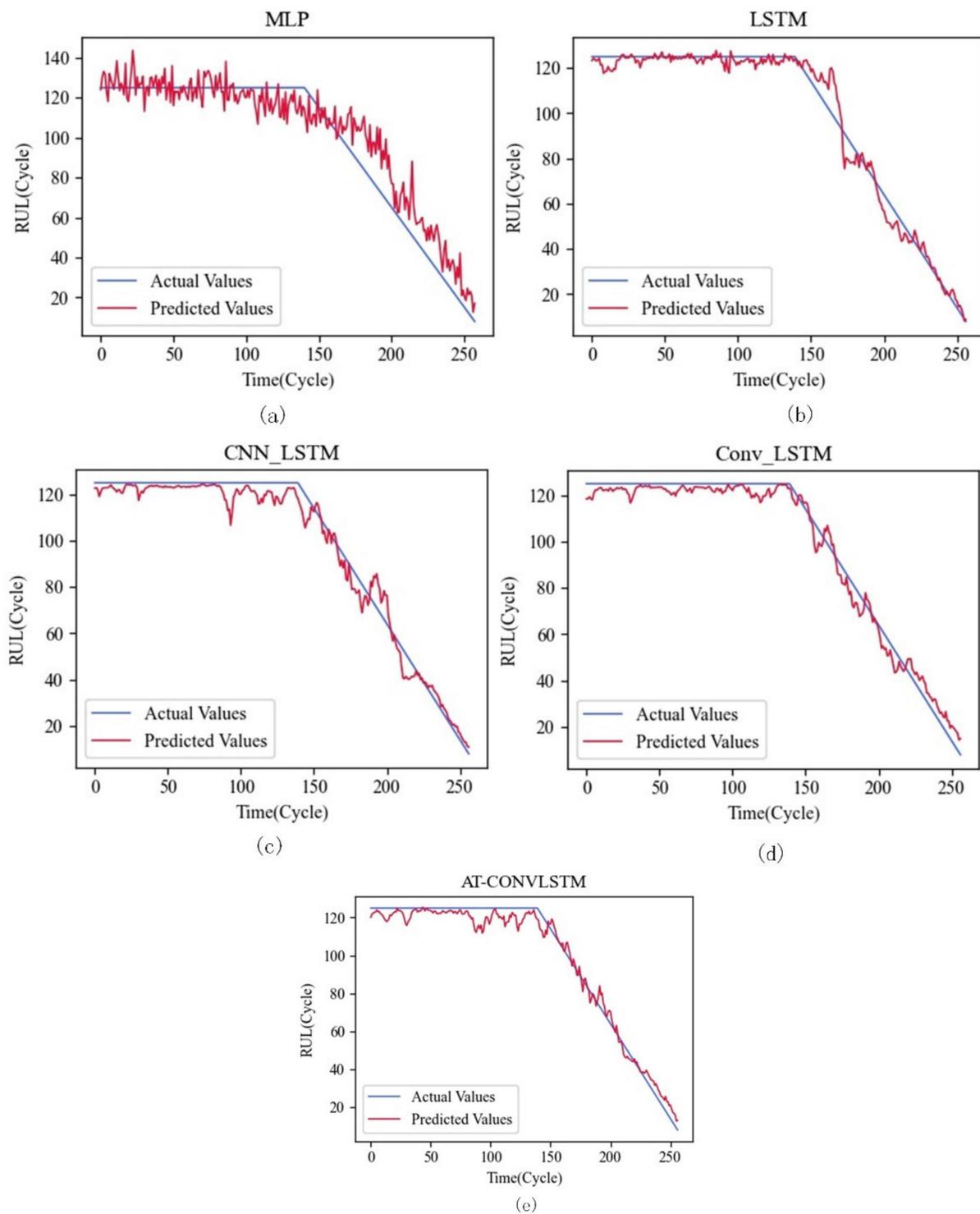| Metrics	MLP	LSTM	CNN-LSTM	Conv-LSTM	AT-CONVLSTM
RMSE $(\times 10^{-3})$	9.68	4.71	5.83	4.60	3.48
MAE $(\times 10^{-3})$	9.37	3.57	4.68	3.45	2.77

Figure 7. RUL prediction results of FD003: (**a**) MLP; (**b**) LSTM; (**c**) CNN-LSTM; (**d**) Conv-LSTM; (**e**) AT-CONVLSTM.

Similarly, Table 5 shows the results of MLP, LSTM, CNN-LSTM, Conv-LSTM, and AT-CONVLSTM network models for SOC prediction of lithium-ion batteries. Similar to the RUL prediction results of the engine, the MLP, LSTM, and CNN-LSTM networks have poor accuracy in the prediction of SOC. Conversely, the Conv-LSTM network can adaptively obtain the coupling information of time and space in the input data and hidden state by using the convolution network, and the effect is better. The proposed AT-CONVLSTM network model applies the attention mechanism to the convolutional long-term and short-

term memory network and uses the attention mechanism to integrate the most relevant information into its prediction based on the contribution of the input features to the SOC, so the prediction results are in good agreement with the actual SOC curve of the lithium-ion battery. The detailed results of predicting the lithium-ion battery test data set using MLP, LSTM, CNN-LSTM, Conv-LSTM, and the AT-CONVLSTM network method proposed in this paper are shown in Figure 8.
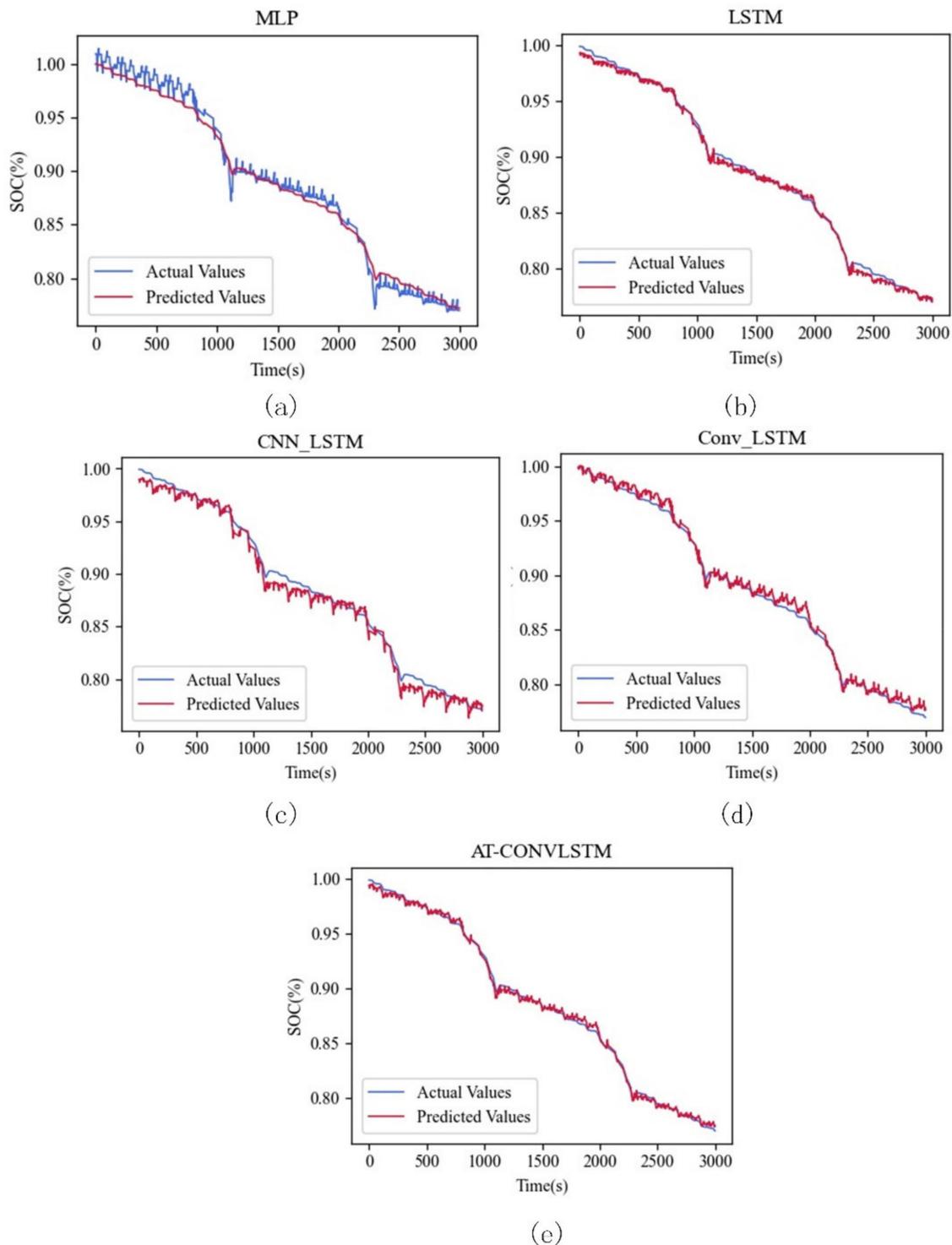

Figure 8. SOC prediction results: (**a**) MLP; (**b**) LSTM; (**c**) CNN-LSTM; (**d**) Conv-LSTM; (**e**) AT-CONVLSTM.

To show the comparison of the prediction results of different algorithms more intuitively, some of the predicted results are shown as follows. The yellow part is the prediction result, and the blue part is the actual result of the proposed method.

5. Conclusions

Aiming at the characteristics of a traditional LSTM recurrent neural network that does not consider the relationship between sequences when predicting time series, this paper improves LSTM and integrates CNN and the attention mechanism into the LSTM network to improve the prediction accuracy of the model. The spatial attention module is utilized first to weigh distinct inputs according to their contribution to the output, and then the processed data is sent to the upgraded Conv-LSTM network. In contrast to standard CNN-LSTM networks, the Conv-LSTM network here takes into account, not just the correlation between the input features of each time step, but also the connection between hidden states. Finally, the proposed AT-CONVLSTM is used to estimate the RUL prediction of the aeroengine and the SOC prediction of a lithium-ion battery of the electric vehicle. The final experimental results are obtained by conducting multiple separate experiments. The specific experimental results are higher that on the RUL verification data set of aeroengine FD001, the RMSE of the proposed algorithm is 76.4%, 33.2%, 14.3%, and 4% lower than that of MLP, LSTM, CNN-LSTM, Conv-LSTM, respectively. In the SOC prediction experiment of the electric vehicle lithium-ion battery, the average RMSE of the proposed algorithm is reduced by 178%, 35%, 67%, and 32 %, respectively, compared with MLP, LSTM, CNN-LSTM, Conv-LSTM. The results of different data sets show that the proposed model is superior to MLP, LSTM, CNN-LSTM, and Conv-LSTM.

References

1. Zhang, X.; Wang, Y.; Yang, D.; Chen, Z. An on-line estimation of battery pack parameters and state-of-charge using dual filters based on pack model. *Energy* **2016**, *115*, 219–229. [CrossRef]
2. Manthiram, A. An outlook on lithium ion battery technology. *ACS Cent. Sci.* **2017**, *3*, 1063–1069. [CrossRef] [PubMed]
3. Lai, X.; He, L.; Wang, S.; Zhou, L.; Zhang, Y.; Sun, T.; Zheng, Y. Co-estimation of state of charge and state of power for lithium-ion batteries based on fractional variable-order model. *J. Clean. Prod.* **2020**, *255*, 120203. [CrossRef]
4. Meng, J.; Ricco, M.; Luo, G.; Swierczynski, M.; Stroe, D.; Stroe, A. An overview and comparison of online implementable SOC estimation methods for lithium-ion battery. *IEEE Trans. Ind. Appl.* **2017**, *54*, 1583–1591. [CrossRef]
5. Lu, L.; Han, X.; Li, J.; Hua, J.; Ouyang, M. A review on the key issues for lithium-ion battery management in electric vehicles. *J. Power Sources* **2013**, *226*, 272–288. [CrossRef]
6. Hannan, M.A.; Lipu, M.S.H.; Hussain, A.; Mohamed, A. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renew. Sustain. Energy Rev.* **2017**, *78*, 834–854. [CrossRef]
7. Plett, G.L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—Part 2: Modeling and identification. *J. Power Sources* **2004**, *134*, 262–276. [CrossRef]
8. Yang, F.; Xing, Y.; Wang, D.; Tsui, K. 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* **2016**, *164*, 387–399. [CrossRef]
9. Ng, K.S.; Moo, C.S.; Chen, Y.P.; Hsieh, Y.C. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Appl. Energy* **2009**, *86*, 1506–1511. [CrossRef]

10. Nejad, S.; Gladwin, D.T.; Stone, D.A. A systematic review of lumped-parameter equivalent circuit models for realtime estimation of lithium-ion battery states. *J. Power Sources* **2016**, *316*, 183–196. [CrossRef]

11. El Din, M.S.; Hussein, A.A.; Abdel-Hafez, M.F. Improved battery SOC estimation accuracy using a modified UKF with an adaptive cell model under real EV operating conditions. *IEEE Trans. Transp. Electrif.* **2018**, *4*, 408–417. [CrossRef]

12. Zou, C.; Manzie, C.; Nesic, D. A framework for simplification of PDE-based lithium-ion battery models. *IEEE Trans. Control. Syst. Technol.* **2016**, *24*, 1594–1609. [CrossRef]

13. Zhang, L.; Li, K.; Du, D.J.; Guo, Y.J.; Fei, M.R.; Yang, Z.L. A Sparse Learning Machine for Real-Time SOC Estimation of Li-ion Batteries. *IEEE Access* **2020**, *8*, 156165–156176. [CrossRef]

14. Li, C.; Chen, Z.; Cui, J.; Wang, Y.; Zou, F. The lithium-ion battery state-of-charge estimation using random forest regression. In Proceedings of the 2014 Prognostics and System Health Management Conference (PHM), Zhangjiajie, China, 24–27 August 2014.

15. Lipu, M.S.H.; Hannan, M.A.; Hussain, A.; Saad, M.H.M. Optimal BP neural network algorithm for state of charge estimation of lithium-ion battery using PSO with PCA feature selection. *J. Renew. Sustain. Energy* **2018**, *9*, 064102. [CrossRef]

16. Chaoui, H.; Ibe-Ekeocha, C.C. State of charge and state of health estimation for lithium batteries using recurrent neural networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 8773–8883. [CrossRef]

17. Yang, F.; Li, W.; Li, C.; Miao, Q. State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network. *Energy* **2019**, *175*, 66–75. [CrossRef]

18. Yang, F.; Song, X.; Xu, F.; Tsui, K.-L. State-of-Charge estimation of lithium-ion batteries via long short-term memory network. *IEEE Access* **2019**, *7*, 53792–53799. [CrossRef]

19. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]

20. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]

21. Srinivasu, P.; SivaSai, J.; Ijaz, M.; Bhoi, A.; Kim, W.; Kang, J. Classification of Skin Disease Using Deep Learning Neural Networks with MobileNet V2 and LSTM. *Sensors* **2021**, *21*, 2852. [CrossRef]

22. Zhao, J.; Mao, X.; Chen, L. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomed. Signal Process. Control* **2019**, *47*, 312–323.

23. Han, X.; Laga, H.; Bennamoun, M. Image-Based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1578–1604. [CrossRef] [PubMed]

24. Qin, Y.; Song, D.; Cheng, H.; Cheng, W.; Jiang, G.; Cottrell, G. A dual-stage attention-based recurrent neural network for time series prediction. In Proceedings of the 2017 26th International Joint Conference on Artificial Intelligence (IJCAI), Melbourne, VIC, Australia, 19–25 August 2017.

25. Zheng, H.; Lin, F.; Feng, X.; Chen, Y. A Hybrid Deep Learning Model with Attention-Based Conv-LSTM Networks for Short-Term Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 6910–6920. [CrossRef]

26. Luo, C.; Li, X.; Ye, Y. PFST-LSTM: A SpatioTemporal LSTM Model with Pseudoflow Prediction for Precipitation Nowcasting. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 843–857. [CrossRef]

27. Tian, H.; Ouyang, B. Estimation of EV battery SOC based on KF dynamic neural network with GA. In Proceedings of the 30th Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018.

28. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In Proceedings of the 2008 International Conference on Prognostics and Health Management (PHM), Denver, CO, USA, 6–9 October 2008.

29. Song, Y.; Gao, S.; Li, Y.; Jia, L.; Li, Q.; Pang, F. Distributed Attention-Based Temporal Convolutional Network for Remaining Useful Life Prediction. *IEEE Internet Things J.* **2021**, *8*, 9594–9602. [CrossRef]