A Data-Driven Auto-CNN-LSTM Prediction Model for Lithium-Ion Battery Remaining Useful Life

Lei Ren, *Member, IEEE*, Jiabao Dong, Xiaokang Wang, *Member, IEEE*, Zihao Meng, Li Zhao, and M. Jamal Deen, *Fellow, IEEE*

***Abstract*—Integration of each aspect of the manufacturing process with the new generation of information technology such as the Internet of Things, big data, and cloud computing makes industrial manufacturing systems more flexible and intelligent. Industrial big data, recording all aspects of the industrial production process, contain the key value for industrial intelligence. For industrial manufacturing, an essential and widely used electronic device is the lithium-ion battery (LIB). However, accurately predicting the remaining useful life (RUL) of LIB is urgently needed to reduce unexpected maintenance and avoid accidents. Due to insufficient amount of degradation data, the prediction accuracy of data-driven methods is greatly limited. Besides, mathematical models established by model-driven methods to represent degradation process are unstable because of external factors like temperature. To solve this problem, a new LIB RUL prediction method based on improved convolution neural network (CNN) and long short-term memory (LSTM), namely Auto-CNN-LSTM, is proposed in this article. This method is developed based on deep CNN and LSTM to mine deeper information in finite data. In this method, an autoencoder is utilized to augment the dimensions of data for more effective training of CNN and LSTM. In order to obtain continuous and stable output, a filter to smooth the predicted value is used. Comparing with other commonly used methods, experiments on a real-world dataset demonstrate the effectiveness of the proposed method.**

Manuscript received March 14, 2020; revised May 18, 2020 and June 22, 2020; accepted July 6, 2020. Date of publication July 9, 2020; date of current version February 22, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1004001 and in part by the National Science Foundation of China (NSFC) under Project 61572057 and Project 61836001. Paper no. TII-20-1335. *(Corresponding author: Lei Ren.)*

Lei Ren, Jiabao Dong, Zihao Meng, and Li Zhao are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China, and also with the Beijing Advanced Innovation Center for Big Data-Based Precision Medicine, Beihang University, Beijing 100191, China (e-mail: renlei@buaa.edu.cn; aarondong@buaa.edu.cn; mzh1996@buaa.edu.cn; zliyjy@163.com).

Xiaokang Wang is with the Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada (e-mail: xkwang@stfx.ca).

M. Jamal Deen is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada (e-mail: jamal@mcmaster.ca).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2020.3008223

***Index Terms*—Autoencoder, convolution neural network (CNN), lithium-ion battery (LIB), long short-term memory (LSTM), remaining useful life (RUL) prediction, industrial artificial intelligence.**

I. Introduction

INTELLIGENT manufacturing combines the new generation of information, computational and communication technologies such as the big data, cloud computing, edge computing, artificial intelligence, and Internet of Things into every aspect of manufacturing to achieve a more intelligent and automated manufacturing process [1]–[3]. For this purpose, intelligent manufacturing big data was selected as the research target [4], [5]. This is because intelligent manufacturing big data records all aspects of the production process including product parameters, production environment, logistics and transportation, and after-sales services [2], [5]. These heterogeneous and complex big data contain hidden valuable information of intelligent manufacturing [1], [4], [5], which bring unprecedented challenges for in-depth understanding of intelligent manufacturing.

At the same time, lithium-ion battery (LIB), as a green high-energy rechargeable electrical power resource of intelligent manufacturing with prominent characteristics such as high capacity, reliability, and safety, is widely used in intelligent manufacturing involving computational engineering, logistics, and aerospace [6]. However, the degradation of LIBs may significantly lead to a decline in the performance of electrical equipment, which increases the cost of unexpected maintenance. As the discharging time increases, battery failure induced by degradation can shorten the LIB service life and even cause serious accidents [7]. Therefore, the remaining useful life (RUL) of the LIB, referred to as the length of time to the end of useful life [8], should be predicted accurately. This is also of great significance to the safety, stability, and cost of intelligent manufacturing.

Therefore, we should analyze the operation data of LIB to predicate its RUL. At present, the current prediction method of LIB mainly includes model-driven methods and data-driven methods. Model-based methods aim to establish mathematical models to represent the performance during degradation process of LIB, such as physics of failure modeling method [9], Kalman filtering method [10], and particle filtering (PF) method [11]. Existing model-based methods mainly seek a mathematical model to fit the degraded trajectory of the LIB, and then exploit the PF method for prediction. For example, a coupled numerical method was proposed in [12] for the life prediction of LIB for electric vehicles. A double exponential model was proposed to approximate the degraded trajectory of the LIB capacity [13] and initialized model parameters with Dempster–Shafer evidence theory, combining with PF. An improved method based on interaction multimodel particle filter was developed [14], combining various capacity models of LIBs. However, there are still two significant shortcomings in the existing models. 1) Due to complex internal electrochemical characteristics, the battery is susceptible to external factors such as temperature and humidity; so establishing a general accurate mathematical or physical model is difficult [15]. 2) Due to the particle degeneracy problems, the filtering method PF brings disadvantages for the RUL prediction accuracy.

The data-driven methods analyze LIB life characteristic parameters [16], [17] such as capacity, current, voltage, and impedance by intelligent algorithm models such as neural network [18], support vector machine (SVM) [19], and Bayesian regression [20]. An adaptive recurrent neural network (RNN) RUL prediction method for LIB impedance spectrum data was proposed in [21]. A method for predicting the remaining life of LIBs based on the nonlinear auto regressive model for its capacity, and that closely approaches its nonlinear degradation trajectory was presented in [22]. In [23], the Gaussian process regression (GPR) method that takes the capacity characteristics of LIBs into consideration was used to describe its uncertainty of the prediction. Even though the shortcomings of model-driven methods are avoided by selecting data-driven methods, there are still limitations. 1) Some methods, such as GPR, require large statistical datasets to complete the learning process of the model. 2) Most methods obtain only a single prediction result without uncertainty information, largely limiting their prediction accuracy.

To predict RUL with satisfactory accuracy, three main challenges have to be addressed. 1) Limitation of the amount of data and noise in data are impeding offline training of data-driven models. Meanwhile, information between the state of LIB in adjacent charge and discharge cycles can be used to train the model. Besides, fluctuations among cycles are barely analyzed in most models. 2) Due to the complex and unstructured characteristics of data, analyzing and processing data of LIB for industrial optimization and scheduling is challenging with most data-driven approaches. 3) Most data-driven methods output a single-valued prediction result corresponding to the number of charge and discharge cycles, and the prediction curves are usually discontinuous.

Considering the limitations of previous research and methods, we propose a data-driven prediction model based on improved convolution neural network (CNN) and long short-term memory (LSTM), namely Auto-CNN-LSTM, for LIB RUL, and our main contributions are summarized as follows. 1) A model composed of autoencoder CNN and LSTM is proposed to predict RUL of LIB. CNN and LSTM use the correlation between adjacent cycles of the battery and exploit features among sequences, which is beneficial for attenuating data noise and enhancing prediction accuracy. 2) An auto autoencoder is introduced to process and reconstruct battery data in time series by augmenting dimensions of data, which is used to extract more valuable information from raw data, preparing for more efficient training afterwards. 3) A postsmoothing method is proposed to solve the discontinuity problem in the prediction results. Filters of different types are compared, allowing us to achieve more accurate results than models without filters.

The rest of this article is organized as follows. Section II describes related works in the field of LIB RUL prediction. Section III describes the methodology proposed, followed by the procedure of the experiments. Section IV presents and discusses results of experiments. Finally, Section V concludes this article.

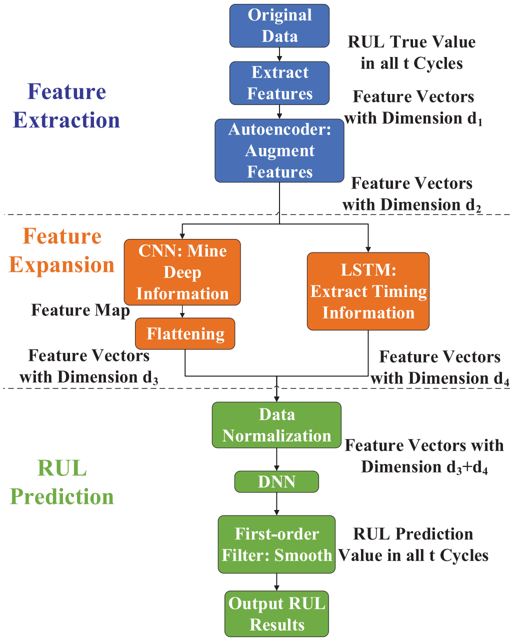

Original Data
RUL True Value in all t Cycles
Extract Features
Feature Extraction
Feature Vectors with Dimension d_1
Autoencoder: Augment Features
Feature Vectors with Dimension d_2
Feature Expansion
CNN: Mine Deep Information
LSTM: Extract Timing Information
Feature Map
Flattening
Feature Vectors with Dimension d_3
Feature Vectors with Dimension d_4
Data Normalization
Feature Vectors with Dimension d_3+d_4
RUL Prediction
DNN
First-order Filter: Smooth
RUL Prediction Value in all t Cycles
Output RUL Results

Fig. 1. Framework for LIB RUL prediction.

II. Methodology

In this section, the framework (Fig. 1) of the proposed LIB RUL prediction method, namely Auto-CNN-LSTM is presented.

Due to the ability of CNN to mine hidden information in limited data and the ability of LSTM to process information in time series, we propose a LIB RUL prediction method based on CNN and LSTM. Moreover, the convolution layers of CNN require a lot of data, but the data of LIB are insufficient. However, by using an autoencoder, we achieved a significant increase in the dimensions of data that can be transmitted to the CNN for more effective training. Also, considering the noise presented in the model prediction curve, we propose a postsmoothing method to reduce noise and solve the discontinuity problem in the prediction results. This improves the interpretability of the prediction results by outputting smooth prediction curve. Algorithm 1 shows the framework of our proposed Auto-CNN-LSTM method.

Algorithm 1: Framework of Auto-CNN-LSTM Proposed by This Article.

```
Input: The set of original feature vectors with same
  dimensions $d_1$, representing each adjacent charging and
  discharging cycle from raw data, $P_n$;
Output: Battery's RUL prediction results corresponding to
  each cycle in the whole process, and the prediction curve.
  1: Vectors with $d_1$ dimensions in $P_n$ are passed through
     the autoencoder, which augments the dimension of
     every vector to be $d_2$;
  2: Feature maps formed by feature vectors with $d_2$
     dimensions are stacked over as feature maps, which
     will be mined for deep information by the CNN;
  3: The same feature maps with $d_2$ dimensions are used to
     extract correlation between adjacent cycles by RNN
     with LSTM;
  4: Features obtained by CNN and LSTM are
     concatenated to be a vector containing information
     both in time series and in depth; then, the vector will
     be operated by DNN with multiple layers to predict
     the final RUL;
  5: The RUL prediction results are filtered to obtain
     smoother and more stable result;
```

In the model, the LIB characteristics are initially extracted from the raw data by the autoencoder, which augments the number of data dimensions from 21 to 50. Then, feature expansion is performed by the CNN and LSTM model. The CNN mines deep information and LSTM extracts the timing information between the data, and the features extracted by CNN and LSTM are combined into the fully connected layer. The RUL prediction of the LIB is output after a seven-layer deep neural network (DNN). Finally, in order to smooth the fitting curve, the output is respectively filtered by first-order linear, second-order, and third-order smoothing networks. Afterwards, the accuracy calculated by these networks is compared, and the best accuracy is selected as the prediction result. Using experiments with real data, the prediction results demonstrated the effectiveness of our proposed method.

A. Autoencoder of Auto-CNN-LSTM

The autoencoder is an unsupervised neural network structure mapping input to output [20]. The network structure of the autoencoder is shown in Fig. 2. It is composed of input, hidden, and output layers. The hidden layer is also referred to as the feature extraction layer. The forward conduction process of the encoder is divided into two parts: encoding and decoding. The encoding function expression is

$$z = E(x) = s(w^1 x + b^1) \tag{1}$$

where $x \in R^{d_1 \times 1}$ denotes the input data, and d_1 denotes the dimension of the input data; $z \in R^{r \times 1}$ denotes the characteristic expression of the hidden layer, and r denotes the number of neurons in the hidden layer; $w^1 \in R^{r \times d_1}$ denotes the input weight of the hidden layer; $b^1 \in R^{r \times 1}$ denotes the implicit input bias; s denotes the activation function.

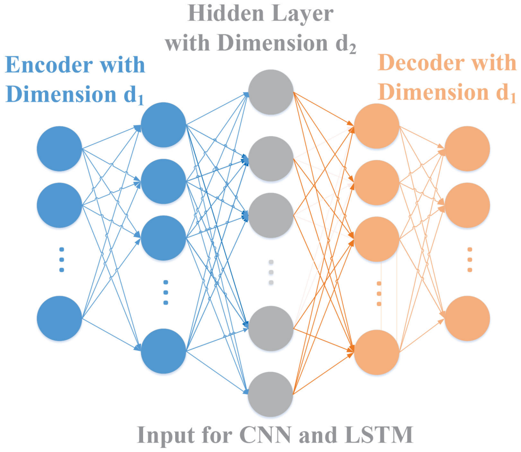

Hidden Layer with Dimension d₂
Encoder with Dimension d₁
Decoder with Dimension d₁
Input for CNN and LSTM

Fig. 2. Autoencoder structure in Auto-CNN-LSTM.

In the decoding process, the feature expression z of the hidden layer is mapped to the input data x through the decoding process, and the function expression $\bar{x}$ is

$$\overline{x} = D(z) = s(w^2 x + b^2) \tag{2}$$

in which $w^2 \in R^{d_1 \times r}$ and $b^2 \in R^{d_1 \times 1}$.

This method applies autoencoder to expand the data due to limitation of battery data when conducting convolution and pooling in CNN. In this way, autoencoder is used to significantly augment the dimensions of the data that is used by the CNN for more effective training.

Vectors with 21 dimensions (d_1) of each adjacent 14 charging and discharging cycles are vertically stacked into a time-domain feature map with size of 14 × 21. The 21-dimensional LIB data vectors are first input to the encoder, then output with 50 dimensions (d_2). After the encoding process of autoencoder, the feature map from the original dataset will be enlarged to a 14 × 50 time-domain feature map, which will be the input of CNN and LSTM.

B. Convolution Neural Network

To the best of our knowledge, this is the first attempt at using CNN for predicting the lifetime of LIBs. A typical CNN consists of convolution layers, pooling layers, and fully connected layers. A discrete two-dimensional convolution formula is

$$A(i,j) = \sum_{m=-s}^{s} \sum_{n=-s}^{s} K(m,n) * z(i+m, j+n). \tag{3}$$

If we assume that the kernel size of matrix K is $(2s+1) \times (2s+1)$, then $A(i, j)$ is the resulting matrix from convolution. Also, $z(i, j)$ denotes the input matrix which is the encoded matrix processed by the encoder.

The CNN module can extract deep information from the feature map output of the autoencoder. According to the results of our extensive tests, we chose the most appropriate CNN structure. Fully connected layers are usually used to map the

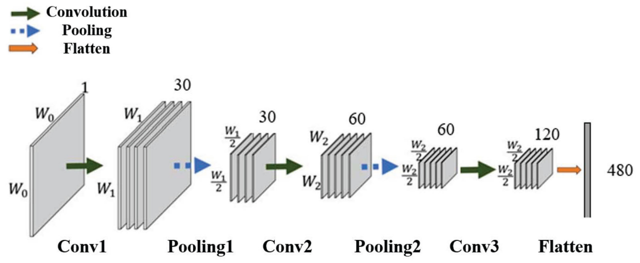


Fig. 3. CNN structure in Auto-CNN-LSTM.

features learned by convolution layers and pooled layers to the sampling space. In our CNN, the fully connected layer is discarded to reduce the amount of calculation, contributing to a more lightweight and effective model without too many parameters to train. The network structure is shown in Fig. 3.

The CNN structure comprises six layers including three convolution layers, two pooling layers, and a flattening layer. The specific network structure is composed in the following order: [Convolution, Max Pooling, Convolution, Max Pooling, Convolution, Flatten]. The CNN utilizes the characteristics of the LIB in each charging and discharging cycle, finding the time-domain characteristics between adjacent charging and discharging cycles that can be extracted by the autoencoder. During the forward propagation of deep CNN, the LIB feature map is gradually blurred, and the global information of each feature map will be gradually highlighted.

Ultimately, information extracted by this section will be transformed into a feature vector containing 480 elements by the flattening layer. This feature vector will be a part of combination with LSTM, to be input into the DNN for final prediction of the RUL.

C. Long Short-Term Memory

Because the data of the LIB are collected in charging and discharging cycles, which are time series, the RNN is suitable to process time series when making use of internal memory. There are both internal feedback connections and feedforward connections between its processing units, which contributes to the processing sequence of information by the RNN. However, since RNN can only store part of the sequence, its performance on a long sequence is worse than that on a short sequence, resulting in a decrease in accuracy for long sequences. In addition, the data sequence of LIBs in this experiment is relatively long and there are hundreds of time series; so LSTM is selected as an alternative to a conventional RNN.

The LSTM is a special kind of RNN. It overcomes the problem that an RNN cannot handle long-distance dependences [24]. There is a certain correlation between the hidden layer nodes in the LSTM. When data are sequentially input into the network, the calculation of the hidden layer nodes depends not only on the input of the current layer but also on the activation values of nodes in the hidden layer in the previous moment. As for the input sequence, the LSTM network layer will work out the hidden layer sequence and the output sequence.

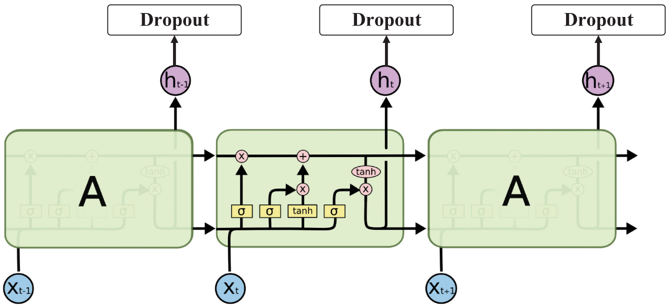


Fig. 4. LSTM structure in Auto-CNN-LSTM.

The RNN module can extract amounts of timing information from the input feature sequences. Building on the work in [25] and our own experience, we chose LSTM to form the RNN module. Part of the LSTM network structure used in our work is shown in Fig. 4.

The RNN structure consists of four layers, including two LSTM layers and two dropout layers. The specific network structure is composed in the following module order: [LSTM, Dropout, LSTM, Dropout]. The structure in Fig. 4 is only half of the whole LSTM module. Although the CNN structure proposed above can find the global characteristics of the feature map, the RNN structure is still preferred, which benefits from its unique memory capability. The calculation formula for each gate function and state transfer process in the LSTM module is as follows:

$$f^t = \sigma(W_f \cdot [h^{t-1}, x^t] + b_f) \tag{4}$$

$$i^t = \sigma(W_i \cdot [h^{t-1}, x^t] + b_i) \tag{5}$$

$$\overline{C^t} = \tanh(W_g \cdot [h^{t-1}, x^t] + b_g) \tag{6}$$

$$o^t = \sigma(W_o \cdot [h^{t-1}, x^t] + b_o) \tag{7}$$

$$C^t = f^t * C^{t-1} + i^t * \overline{C^t} \tag{8}$$

$$h^t = o^t \cdot \tanh(C^t) \tag{9}$$

in which *f* denotes forgetting gate output, *i* denotes input gate output, *o* denotes output gate output, h^t denotes *t* moment hidden layer output, x^t denotes input at time t, *W* denotes connection weight parameter, *B* denotes offset parameter, and C^t denotes the intermediate variable, which is multiplied by the input gate result and sent to the state space.

With 14 adjacent charging and discharging cycles vertically stacked into a time-domain feature map, the input of the LSTM module is also composed of a sequence of time-domain features of these cycles. In this sequence, each element corresponds to the time-domain characteristics obtained by autoencoder training in one specific charging and discharging cycle.

The RNN can accurately capture the trend of each feature over time obtained by the autoencoder. We explore the use of the LSTM network structure to extract the time-domain characteristics of LIBs and incorporate the extracted features into the feature vectors for the next stage. In the forward propagation process of the LSTM, the characteristics of the LIB output by

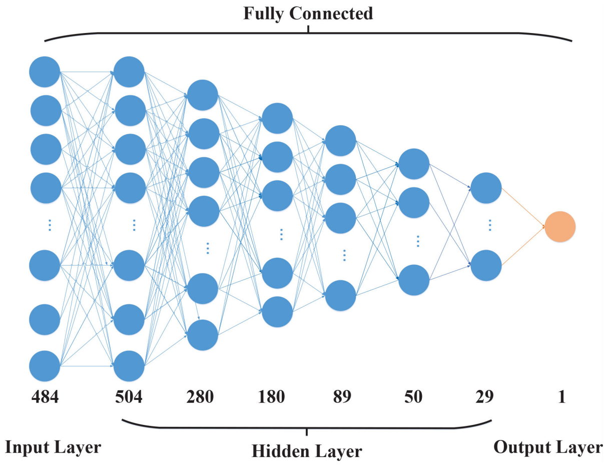


Fig. 5. DNN structure in Auto-CNN-LSTM.

the autoencoder are constructed into a sequence of features in time sequence. After calculation of the network, the correlation between moments in the input sequence will be gradually highlighted.

D. Deep Neural Network

Due to its deep architecture, a DNN model consists of multiple network layers, which can be used to obtain features of the raw data from linear and nonlinear operations [26]. With features containing sequential information obtained by CNN and LSTM, the DNN is capable of predicting RUL with better accuracy.

CNN and LSTM modules obtain and represent features with a 484-dimensional vector from autoencoder. In order to establish an accurate mapping of the above eigenvectors to the battery RUL prediction, this method utilizes a seven-layer DNN to build the final LIB RUL regression model shown in Fig. 5. In this seven-layer DNN structure, the input is a 484-dimensional feature vector, and the output is a 1-D battery life prediction vector ranging from 0 to 1. Besides, in order to establish the LIB RUL regression model with generalization properties, the Dropout and L2 regularization techniques are used in the DNN structure.

It is noted that the root mean square error (RMSE) between the output value of the DNN network and the actual value of the battery RUL is applied to measure the accuracy of prediction results

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum\nolimits_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{10}$$

in which y_i represents RUL prediction result corresponding to the ith moment in a total of m moments, and $\hat{y}_i$ represents true value at the same moment.

Through the back propagation algorithm, the weights of CNN, LSTM, and DNN are updated to realize the network learning function. The Adam optimizer is selected for network training so that the network parameters can quickly converge to an acceptable range. After calculation of the DNN, the output from this part is a 1-D RUL prediction vector which needs to be smoothed.

TABLE I
NOTATIONS APPEARING IN METHODOLOGY

Model	Notation	Meaning
Autoencoder	x $z = E(x)$ $\overline{x} = D(z)$	Input Feature Map for Encoder Encoded Map output by Encoder Feature Map output by Decoder
CNN	A	Convolution Result Operated on z
LSTM	f^t i^t $\overline{C^t}$ o^t h^t	Forgetting Gate Output Input Gate Output Intermediate Variable Input Gate Output Hidden Layer Output

E. Smoothing Method in Auto-CNN-LSTM

After directly being processed by the LIB RUL regression model composed of the DNN described above, the output is often discontinuous, unlike true remaining life of the battery. In view of this problem, we attempt to smooth the predicted value by passing the output of the prediction model through a filter for a more continuous and stable output. The method we select to smooth the curve is the least squares polynomial approximation.

In order to explore the influence of filters with different orders, we studied the results of predictor with no filtering and by using a linear filter, a second-order filter, and a third-order filter. According to the analysis of the prediction errors in these four cases, the output of the first-order filter is selected as the final predicted value of the LIB RUL. This choice meets the basic assumption that a general battery's RUL value should be linear with operating time.

The first-order linear filter predicts the RUL of the LIB at time t, which depends not only on the output of the DNN LIB regression model at time t but also at previous moments at t–7, t–6, t–5, and t–4 and output effects at time t–3, t–2, and t–1. Experimental analysis suggests that the predicted output values by the first-order filter and the predicted output values directly without any filter have almost no difference in accuracy, but the filtered curve is smoother and more stable. All the notations appearing in this section are listed in Table I.

III. Experiments

In this section, experiments will be carried out to demonstrate the performance of the proposed model for LIBs RUL, in which the #5, #6, and #25–#27 batteries' dataset of the NASA PcoE will be selected as the training dataset and the #7 and #28 batteries' dataset will be selected as the testing dataset.

A. Data Description

In this section, as shown in Fig. 6, we can obtain the data via the following charging and discharging processes.

1) The first step is the charging process. The LIB will be charged by a constant current (CC) at 1.5 A until its voltage equals 4.2 V. Then, the LIB will be maintained at a constant voltage at 4.2 V until the charging current decreases to 20 mA.

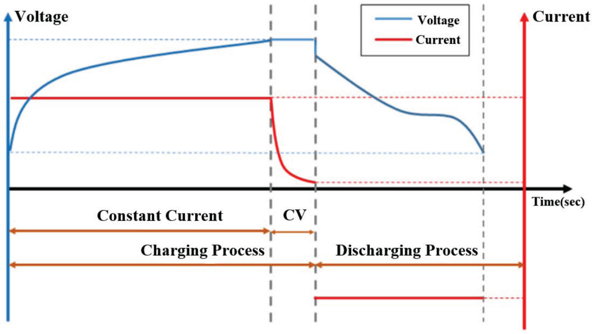


Fig. 6. Charging and discharging process in LIB.

2) The second step is the discharging process in which different LIBs will be discharged with a CC of 2 A until the voltage decreases to different values.

The experiments need to constantly repeat the charging and discharging cycles above to accelerate the battery degradation process. The experiment will end when the LIB life ends, that is, its capacity drops to 70%.

According to data of these kinds of LIB, we took the measured voltage or current by observing how it changes to some specified value. For instance, we took the value of the largest measured voltage in every complete charging process. Meanwhile, we record the time and value when the charging (discharging) current (voltage) gets its preset value described in the dataset document. In this way, a 21-dimensional vector is formed to include physical variants such as voltage measured at charge and temperature at discharge for every charging and discharging cycle. There are 168 cycles for #5–#7 LIBs and 28 cycles for #25–#28 LIBs.

B. Feature Extraction by Autoencoder

After the characteristics of each LIB are extracted as described above, this method applies autoencoder to augment the dimension of raw data, for LIBs #5–#7 and #25–#28. Vectors with 21 dimensions of each of the original adjacent 14 charging and discharging cycles are vertically stacked into a time-domain feature map with a size of 14 × 21. The 21-dimensional LIB data vectors are input to the encoder to get an output with 50 dimensions. After the encoding process of autoencoder, the feature maps from original dataset will be enlarged to 14 × 50 time-domain feature maps, which will be the input of the CNN and the LSTM.

Note that the augmented number of dimension was determined with a set of controlled experiments. In this set of experiments, the number of output dimensions was selected as 40, 45, 50, 55, and 60, while other conditions were kept the same. The final RUL prediction accuracy with different output dimensions of the encoder is compared as follows. According to the comparison results in Table II, the dimension number of encoder is selected to be 50, which means the feature maps input for CNN and LSTM are 14 × 50.

TABLE II
COMPARISON OF ACCURACY WITH DIFFERENT OUTPUT SIZES OF ENCODER

Output Size of Encoder	Filters to Smooth	*RMSE(%)*	*Accuracy(%)*
40	without filter	9.43	90.57
	Linear filter	9.32	91.68
	Second-order filter	9.39	90.61
	Third-order filter	9.39	90.61
45	without filter	8.04	91.96
	Linear filter	7.98	92.02
	Second-order filter	8.03	91.97
	Third-order filter	8.03	91.97
50	**without filter**	**5.03**	**94.97**
	Linear filter	**4.84**	**95.16**
	Second-order filter	**4.98**	**95.02**
	Third-order filter	**4.98**	**95.02**
55	without filter	7.12	92.88
	Linear filter	6.97	93.03
	Second-order filter	7.09	92.81
	Third-order filter	7.09	92.81
60	without filter	7.91	92.09
	Linear filter	7.77	92.23
	Second-order filter	7.89	92.11
	Third-order filter	7.89	92.11

The encoder outputting dimensions are set to vary from 40 to 60 and the filters are selected as linear, second-order, and third-order.

C. Feature Fusion for CNN and LSTM

In the forward propagation process of the LSTM, the characteristics of the LIB output by the autoencoder are constructed into a sequence of features in time sequence. The CNN module extracts deep information from the feature maps with size of 14 × 50 output from the autoencoder. During the forward propagation of the deep CNN, the LIB feature map is gradually blurred, but the global information of each feature map or correlation between adjacent moments will be gradually highlighted. Ultimately, information extracted by CNN and LSTM will be concatenated to be a feature vector containing 484 elements by the flattening layer, and this feature vector will be input into the DNN for the final RUL prediction before smoothing.

D. Data Normalization

In the feature vectors, different physical variants possess different units such as ampere, volt, and degree celsius. The value of these variants can also be extremely different so that their weights on the final result vary a lot and reduces the training efficiency. Consequently, data normalization is used in our model to unify all features to the same scale, which helps every element predicting the RUL with the same weight. We transformed the range of the compressed features to [0,1] with the minimum–maximum normalization method, balancing the predicting weight for every variant with different magnitudes.

E. Training of Auto-CNN-LSTM Model

For autoencoder, there are three layers in which rectified linear unit (ReLu) and sigmoid function are, respectively, utilized as the activation function of the hidden layer and output layer.

For the CNN, its structure in this work is composed in the following order: [Convolution, Max Pooling, Convolution, Max

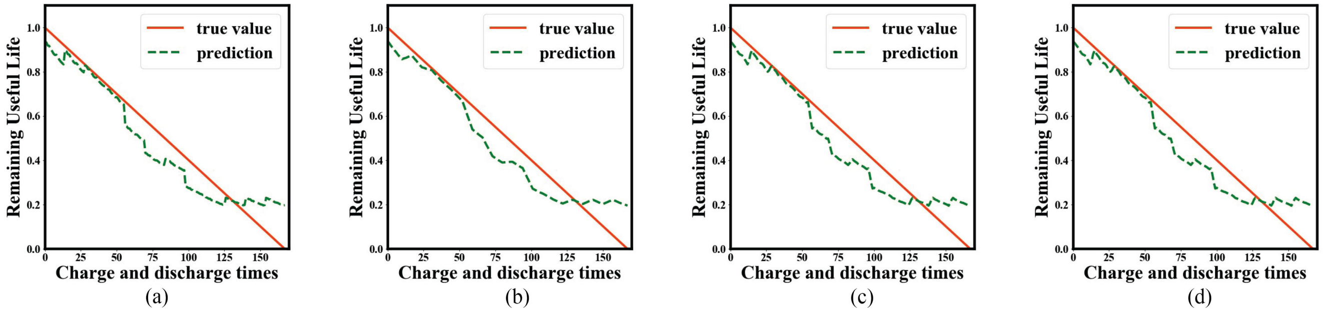


Fig. 7. Prediction results of Auto-CNN-LSTM with filters for #7 LIB. (a) without filter. (b) first-order linear filter. (c) second-order filter. (d) third-order filter.

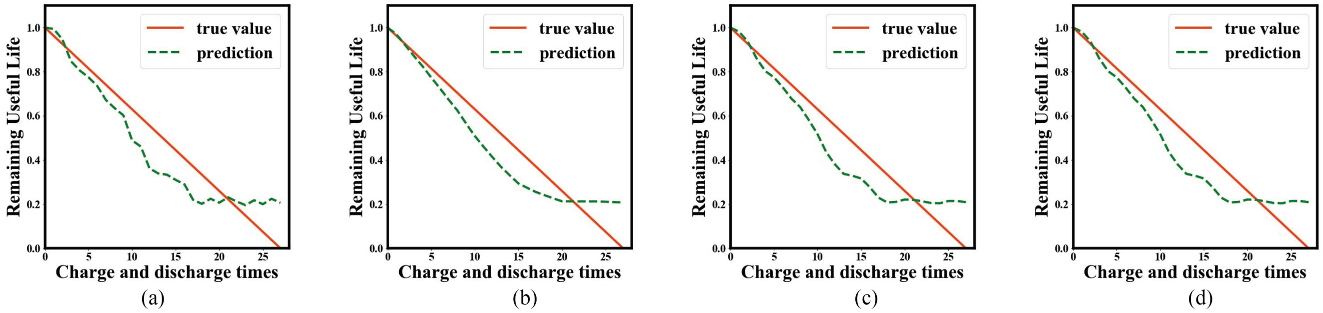


Fig. 8. Prediction results of Auto-CNN-LSTM with filters for #28 LIB. (a) without filter. (b) first-order linear filter. (c) second-order filter. (d) third-order filter.

Pooling, Convolution, Flatten]. The kernel size of the convolution layer is finally determined as [(3 × 3), (3 × 3), (2 × 2)], which can extract features among adjacent physical variables in a vector horizontally and extract timing features among adjacent vectors vertically. At the same time, the largest size of the pooling layers is determined to be 2 × 2. Each layer selected ReLu as the activation function

$$f(x) = \max(0, x). \tag{11}$$

The LSTM structure consists of four layers, including two LSTM layers and two dropout layers in such sequence: [LSTM, Dropout, LSTM, Dropout].

The DNN parameters were trained with normalized training data. In order to establish an accurate mapping of the above eigenvectors to the battery RUL prediction, a seven-layer DNN is used for the final LIB RUL regression prediction. In this seven-layer DNN structure, the input of the DNN model is a concatenated feature vector extracted by CNN and LSTM, and the output is the final RUL prediction vector ranging from 0 to 1. Also, in order to enable the established LIB RUL regression model with generalization properties, the dropout and L2 regularization techniques are used in the DNN structure. The number of neurons in each layer is determined as [504, 280, 180, 89, 50, 29, 1] according to extensive experiments. The parameters of Dropout are determined to be [0.59, 0.59, 0.589, 0.591, 0.59, 0.59], and the coefficient of L2 regularization is 0.001. The activation function of each layer selects ReLU. After training, the DNN model outputs RUL of the LIB while the input is from the combination of CNN and LSTM.

F. Prediction and Evaluation of RUL

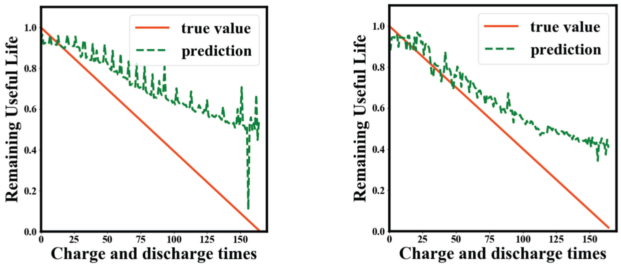
To measure the performance of the proposed model, the dataset of #7 and #28 LIB will be used as the testing dataset. We compared the performance of proposed model with two other models, which are autoencoder-DNN (ADNN) and SVM.

IV. Results and Discussion

A. Analysis of Results

After experiments based on the process described above, we got the results of our model. Figs. 7 and 8 show the results of a #7 and #28 battery prediction with Auto-CNN-LSTM. As shown in Figs. 7 and 8, there are four different subfigures. These subfigures show the charge and discharge cycles in RUL of LIB, and versus the the normalized values of remaining cycle times for LIB without a filter [(a)] and with three types of filters [(b)–(d)]. The orange solid lines are used to show the true values of the relationship between RUL and charge and discharge times, and the green dashed lines show the predicted values.

From the experimental results, the RUL prediction results are satisfactory. The RMSE is 5.03%, and the accuracy is 94.97%. The reason for the error may be that the amount of data is still insufficient even though the original data is augmented with the autoencoder. Another reason may be the effect of noise in



(a) (b)

Fig. 9. Prediction results of ADNN and SVM for #7 LIB. (a) ADNN. (b) SVM.

the data. Although the autoencoder is added for noise reduction, the effect of data noise cannot be completely avoided. As shown in Figs. 7(a) and 8(a), the RUL prediction curves are well-fitted, but the curve at 60th–100th generation for #7 LIB (15th–20th for #28 LIB) on the abscissa is still rough. Therefore, this article adopts the idea of filtering and performs first-order linear, second-order, and third-order filtering.

B. Analysis of Results With Filters

To explore the influence of different order filters on the prediction model output, Fig. 7(b)–(d) shows the results obtained by output through the first-order linear, second-order, and third-order filters. In terms of RMSE, corresponding value after filter in sequence of the three filters above are 4.84%, 4.98%, and 4.98%. By comparing prediction errors in four cases, the output of the first-order filter is finally selected as the final predicted value of the LIB's RUL. This also meets the basic assumption that the general battery RUL value is linear with operating time. In the case of first-order filtering, the accuracy rate reaches a maximum of 95.16%. Compared with Fig. 7(a) without filtering, the curve of Fig. 7(b) is also smoother.

C. Comparison With Other Models

In order to evaluate the performance of our model, we have compared experimental results with the other two methods [27], [28] using the same training and testing sets. As shown in Fig. 9(a), the prediction of #7 LIB with ADNN model proposed in [27] is compared to the true value. The ADNN model is a data-driven method based on an autoencoder and DNN. In Fig. 9(a), the horizontal axis represents charge and discharge cycles in RUL of LIB, and the vertical axis represents normalized values of remaining cycle times for LIB. In this figure, the orange-solid line shows the true value, and the green-dashed line shows the predicted values using ADNN. From the curves shown in Fig. 9(a), we could find that the RUL prediction curve has a satisfactory performance with an RMSE of 11.80% and an accuracy rate of 88.20%. However, a simple fully connected neural network cannot give good results, and it is 6.96% less accurate than Auto-CNN-LSTM proposed in the current work described in this article. Next, Fig. 9(b) shows the prediction of #7 LIB with an SVM model described in [28]. From the experimental results in Table III, the RMSE of SVM is 18.23% and the accuracy rate is 81.77%. From these two comparisons using ADNN and SVM, it is shown that such traditional data-driven models are not ideal for LIB RUL prediction.

TABLE III
COMPARISON OF EXPERIMENT RESULTS WITH TESTING DATA

Method	*RMSE(%)*	*Accuracy(%)*
Auto-CNN-LSTM	5.03	94.97
Linear Filter	**4.84**	**95.16**
Second-order Filter	4.98	95.02
Third-order Filter	4.98	95.02
ADNN	11.80	88.20
SVM	18.23	81.77

As shown in Table III, the comparison results demonstrate that Auto-CNN-LSTM gives better accuracy than ADNN or SVM for LIB RUL prediction. After filtering, the prediction accuracy is further improved and the result of linear filtering is the best. Also, note that the RMSE of the experimental results in the training data is 3.52%, which is slightly better than the same model performed with test data because of overfitting. Overfitting may result in noncompletely optimized parameters or relatively a small L2 regularization coefficient.

V. CONCLUSION

Currently, we are witnessing significant advances in intelligent manufacturing due to the integration of information, communication, big data, and computing technologies in the manufacturing process. For intelligent manufacturing systems, efficient and long lifetime energy sources such as LIBs are essential. However, an important problem with LIBs is how to accurately predict their RUL.

In this article, we proposed a three-step data-driven process-feature extractor, feature expansion, and RUL prediction to determine the RUL of LIBs. Our proposed method, called the Auto-CNN-LSTM prediction, was an autoencoder to augment the original data which are then mined for deep information with a CNN to get the feature map. In parallel, we used LSTM to get its time series which are combined with the feature map and normalized. Then a 7-layer DNN was used to predict the RUL of the LIB and the predicted result was smoothed using a first-order filter. Using the proposed Auto-CNN-LSTM model, and training with true datasets, the predicted results had an RMSE of 4.8%, significantly outperforming two other data-driven models based on ADNN (with error of 11.8%) and SVM (with error of 18.2%). In the future, we plan to investigate other techniques to further improve the RUL prediction accuracy by optimizing the number of neural network layers, dropout, and L2 regularization parameters as well as applying our model to other prediction problems where lifetime in time series is important.

REFERENCES

[1] L. Ren, Y. Laili, X. Li, and X. Wang, "Coding-based large-scale task assignment for industrial edge intelligence," *IEEE Trans. Netw. Sci. Eng.*, to be published. doi: 10.1109/TNSE.2019.2942042.

[2] L. Ren, Z. Meng, X. Wang, R. Lu, and L. T. Yang, "A wide-deep-sequence model based quality prediction method in industrial process analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published. doi: 10.1109/TNNLS.2020.3001602.

[3] X. Wang, L. T. Yang, L. Song, H. Wang, L. Ren, and M. J. Deen, "A tensor-based multi-attributes visual feature recognition method for industrial intelligence," *IEEE Trans. Ind. Informat*, to be published. doi: 10.1109/TII.2020.2999901.

[4] X. Wang, L. T. Yang, Y. Wang, L. Ren, and M. J. Deen, "ADTT: A highly-efficient distributed tensor-train decomposition method for IIoT big data," *IEEE Trans. Ind. Informat*, to be published. doi: 10.1109/TII.2020.2967768.

[5] L. Ren, Z. Meng, X. Wang, L. Zhang, and L. T. Yang, "A data-driven approach of product quality prediction for complex production systems," *IEEE Trans. Ind. Informat*, to be published. doi: 10.1109/TII.2020.3001054.

[6] H. Li, K. Ota, and M. Dong, "Energy cooperation in battery-free wireless communications with radio frequency energy harvesting," *ACM Trans. Embedded Comput. Syst.*, vol. 17, no. 2, 2018, Art. no. 44.

[7] S. Yuchen, L. Datong, H. Yandong, Y. Jinxiang, and P. Yu, "Satellite lithium-ion battery remaining useful life estimation with an iterative updated RVM fused with the KF algorithm," *Chin. J. Aeronaut.*, vol. 31, no. 1, pp. 31–40, 2018.

[8] D. Wang, J. Liu, and R. Srinivasan, "Data-driven soft sensor approach for quality prediction in a refining process," *IEEE Trans. Ind. Informat*, vol. 6, no. 1, pp. 11–17, Feb. 2010.

[9] A. T. Shimamoto, B. R. Tanaka, and C. K. Tanaka, "A study on evaluation method for the lithium-ion battery life performance for stationary use," in *Proc. Int. Conf. Clean Elect. Power*, Alghero, Italy, Jun. 11–13, 2013, pp. 115–119.

[10] S. Lee, J. Kim, J. Lee, and B. H. Cho, "State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge," *J. Power Sources*, vol. 185, no. 2, pp. 1367–1373, 2008.

[11] R. Xiong, Y. Zhang, H. He, X. Zhou, and M. G. Pecht, "A double-scale, particle-filtering, energy state prediction algorithm for lithium-ion batteries," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1526–1538, Feb. 2018.

[12] B. Saha, K. Goebel, S. Poll, and J. Christophersen, "Prognostics methods for battery health monitoring using a Bayesian framework," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 2, pp. 291–296, Feb. 2009.

[13] W. He, N. Williard, M. Osterman, and M. Pecht, "Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method," *J. Power Sources*, vol. 196, no. 23, pp. 10314–10321, 2011.

[14] X. Su, S. Wang, M. Pecht, L. Zhao, and Z. Ye, "Interacting multiple model particle filter for prognostics of lithium-ion batteries," *Microelectron. Rel.*, vol. 70, pp. 59–69, 2017.

[15] W. Waag, C. Fleischer, and D. U. Sauer, "Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles," *J. Power Sources*, vol. 258, pp. 321–339, 2014.

[16] C. Zhang, M. Dong, and K. Ota, "Enabling computational intelligence for green Internet of Things: Data-driven adaptation in LPWA networking," *IEEE Comput. Intell. Mag.*, vol. 15, no. 1, pp. 32–43, Feb. 2020.

[17] X. Liu *et al.*, "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *J. Parallel Distrib. Comput.*, vol. 135, pp. 140–155, 2020.

[18] H.-T. Lin, T.-J. Liang, and S.-M. Chen, "Estimation of battery state of health using probabilistic neural network," *IEEE Trans. Ind. Informat*, vol. 9, no. 2, pp. 679–685, May 2013.

[19] X. Chen, J. Yu, D. Tang, and Y. Wang, "Probabilistic residual life prediction for lithium-ion batteries based on Bayesian LS-SVR," *Acta Aeronaut. Astronaut. Sin.*, vol. 34, no. 9, pp. 2219–2229, 2013.

[20] M. Mishra, J. Martinsson, M. Rantatalo, and K. Goebel, "Bayesian hierarchical model-based prognostics for lithium-ion batteries," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 25–35, 2018.

[21] M. Pecht, "A prognostics and health management roadmap for information and electronics-rich systems," *IEICE ESS Fund. Rev.*, vol. 3, no. 4, pp. 19–30, 2009.

[22] D. Liu, Y. Luo, Y. Peng, X. Peng, and M. Pecht, "Lithium-ion battery remaining useful life estimation based on nonlinear AR model combined with degradation feature," in *Proc. Annu. Conf. Prognostics Health Manag. Soc.*, Denver, USA, Jun.18–21, 2012, pp. 1803–1836.

[23] K. Goebel, B. Saha, A. Saxena, J. R. Celaya, and J. P. Christophersen, "Prognostics in battery health management," *IEEE Instrum. Meas. Mag.*, vol. 11, no. 4, pp. 33–40, Aug. 2008.

[24] Y. Zuo, Y. Wu, G. Min, C. Huang, and K. Pei, "An intelligent anomaly detection scheme for micro-services architectures with temporal and spatial data analysis," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 2, pp. 548–561, Jun. 2020.

[25] Y. Zuo, Y. Wu, G. Min, and L. Cui, "Learning-based network path planning for traffic engineering," *Future Gener. Comput. Syst.*, vol. 92, pp. 59–67, 2019.

[26] X. Yuan, B. Huang, Y. Wang, C. Yang, and W. Gui, "Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE," *IEEE Trans. Ind. Informat*, vol. 14, no. 7, pp. 3235–3243, Jul. 2018.

[27] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, and L. Zhang, "Remaining useful life prediction for lithium-ion battery: A deep learning approach," *IEEE Access*, vol. 6, pp. 50 587–50 598, Jul. 2018.

[28] J. Wei, G. Dong, and Z. Chen, "Remaining useful life prediction and state of health diagnosis for lithium-ion batteries using particle filter and support vector regression," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5634–5643, Jul. 2018.

Lei Ren (Member, IEEE) received the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2009.

Currently, he is an Associate Professor and the Deputy Head with the Cloud Manufacturing Research Center, School of Automation Science and Electrical Engineering, Beihang University, China. He has authored or co-authored over 60 papers and got nearly 4000 citations according to Google Scholar. His current research interests include cloud manufacturing, industrial Internet-of-Things, AI, and big data.

Jiabao Dong is currently working toward the Ph.D. degree in industrial artificial intelligence at the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.

His current research interests include industry intelligence, robot grasping policy, and semantic segmentation.

Xiaokang Wang (Member, IEEE) received the Ph.D. degree in computer system architecture from the Huazhong University of Science and Technology, Wuhan, China, in 2017.

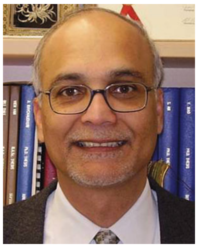
Currently, he is a Postdoctoral Fellow with the Department of Computer Science, St. Francis Xavier University, Antigonish, Canada. His research interests are cyber–physical–social systems, big data, parallel and distributed computing, and cloud-edge computing.

Zihao Meng is currently working toward the Ph.D. degree in industrial artificial intelligence at the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.

His research interests include industry intelligence, natural language processing, and soft sensors.

Li Zhao is currently working toward the graduate degree in industrial artificial intelligence at the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.

His research interests include deep learning, data-driven methods, and remaining useful life prediction of lithium-ion battery.



M. Jamal Deen (Fellow, IEEE) received the Ph.D. degree in electrical engineering and applied physics from Case Western Reserve University, Cleveland, USA, in 1985.

Currently, he is a Distinguished University Professor (highest rank of a Professor in Canada), a Senior Canada Research Chair in information technology, and the Director of the Micro- and Nano-Systems Laboratory, McMaster University. He served as the elected President of the Academy of Science, The Royal Society of Canada, from 2015 to 2017. His current research interests are nanoelectronics, optoelectronics, nanotechnology, data analytics, and their emerging applications to health and environmental sciences. His research record includes more than 600 peer-reviewed articles (about 20% are invited), two textbooks on “Silicon Photonics – Fundamentals and Devices” and “Fiber Optic Communications: Fundamentals and Applications”, 12 awarded patents of which six were extensively used in industry, and twenty-one best paper/poster/presentation awards.

Prof. Deen’s awards and honors include the Callinan Award as well as the Electronics and Photonics Award from the Electrochemical Society, a Humboldt Research Award from the Alexander von Humboldt Foundation, the Eadie Medal from the Royal Society of Canada, McNaughton Gold Medal (highest award for engineers), the Fessenden Medal, and the Ham Education Medal (highest award for educators), all from IEEE Canada. In addition, he was awarded the four honorary doctorate degrees in recognition of his exceptional research and scholarly accomplishments, professionalism, and service. He has also been elected Fellow status in twelve national academies and professional societies including The Royal Society of Canada – The Academies of Arts, Humanities and Sciences (the highest honor for academics, scholars, and artists in Canada), the Canadian Academy of Engineering, Chinese Academy of Sciences, IEEE, APS (American Physical Society), and ECS (Electrochemical Society). Most recently, he was elected to the Order of Canada (July 2018), the highest civilian honor awarded by the Government of Canada.