

Multivariate gated recurrent unit for battery remaining useful life prediction: A deep learning approach

Reza Rouhi Ardeshiri  | Chengbin Ma 

University of Michigan-Shanghai Jiao
Tong University Joint Institute, Shanghai
Jiao Tong University, Shanghai, China

Correspondence

Chengbin Ma, University of Michigan-
Shanghai Jiao Tong University Joint
Institute, Shanghai Jiao Tong University,
Shanghai, China.

Email: chbma@sjtu.edu.cn

Summary

This paper proposes the gated recurrent unit (GRU)-recurrent neural network (RNN), a deep learning approach to predict the remaining useful life (RUL) of lithium-ion batteries (LIBs), accurately. The GRU-RNN structure can self-learn the network parameters utilizing adaptive gradient descent algorithms, leading to a reduced computational cost. Unlike the long short-term memory (LSTM) model, GRU-RNN allows time-series dependencies to be tracked between degraded capacities without using any memory cell. This enables the method to predict non-linear capacity degradations and build an explicitly capacity-oriented RUL predictor. Additionally, feature selection based on the random forest technique was used to enhance the prediction precision. The analyses were conducted based on four separate cycling life testing datasets of a lithium-ion battery. The experimental results indicate that the average percentage of root mean square error for the proposed method is about 2% which respectively is 1.34 times and 8.32 times superior to the LSTM and support vector machine methods. The outcome of this work can be used for managing the Li-ion battery's improvement and optimization.

KEYWORDS

feature engineering, gated recurrent unit, lithium-ion battery, multivariate time series, remaining useful life

1 | INTRODUCTION

Due to the low self-discharge rate of lithium-ion batteries (LiBs), high energy density, and high working voltage, they become a primary choice of the onboard energy storage system in electric vehicles (EVs).^{1,2} Although, in consumer electronics, mobile devices, and EVs, numerous field failures of LiBs are documented. Battery safety has become a critical problem that needs to be intensively investigated. For complicated operational environments in EVs, maintaining battery packs' stability and protection pose a significant technical challenge. Overheat, overcharge, and short circuits are the principal potential failures in the battery packs in EVs.³ This happens due to

several operating conditions, chemical reactions, and mechanical stress.

Prognostics and health management (PHM) of battery technologies has recently attracted a lot of research interest. The enabling discipline of PHM includes methods as well as technologies for assessing the systems' reliability under real-cycle conditions, for the diagnosis of initiated failures and likely failure prognosis.⁴ The lithium-ion battery PHM helps users to make tentative maintenance choices to prevent unexpected failure. As one of the most important states to be tracked in a battery and one of the key approaches for PHM, remaining useful life (RUL) is defined as the remaining number of charge-discharge cycles of the battery before the capacity deteriorates to a

predetermined failure threshold.⁵ Hence, the battery's RUL monitors the future operating status of the battery to manage the charge-discharge of the battery, prolong the battery life, prevent security risks, and decrease the use costs.⁶

1.1 | Literature review

Battery degradation process models are often constructed through traditional prediction of battery RUL approaches based on empirical models or linear model assumptions known as model-driven techniques, which are intended to develop mathematical or physical models for explaining battery degradation mechanism and modify model parameters by employing actual data calculated. For instance, Chen et al proposed a combination of the particle filter (PF) and sliding-window grey model to build a new structure for battery RUL prediction. The proposed method was able to continuously and effectively update model parameters to reflect the changing trend of capacity.⁷ In another work, Chang et al presented a hybrid model in which the unscented Kalman filter method was adopted to achieve a prognostic result based on an estimated model and generate a raw error series.⁸ However, with a complex nonlinear system, an exact mathematical or physical model is impossible to be constructed. In this context, the effectiveness of data-driven models has been attracted high interest. These models typically make decisions regarding online cloud or edge terminal data based on considerable historical data.^{9,10} Such models generally create particular machine learning-based models with more excellent capability for complex nonlinear interactions.

Thus, data-driven methods usually provide more statistical capabilities for data distributions.¹¹ This method builds upon artificial neural networks (ANNs) and can be supervised, semi-supervised, or unsupervised. It uses non-linear functions to establish a relationship between the input and target parameters and uses specific methods to calculate the function parameters. However, the noise and uncleanness of the data dramatically impact solving problems with deep learning methods. Nevertheless, clean data cannot be obtained without noise for data-driven procedures, and it is challenging to identify robust features among the wide range of features such as time domain, time-frequency domain, and frequency domain. Consequently, new feature selection methods such as random forest (RF), normalization cross-correlation indicator method, isometric mapping method, and other techniques are suggested to model the degradation process to achieve a more discriminative feature space. In this regard, a comprehensive paper¹² has been published, which proposed a framework based on

RF classification for lithium-ion battery feature analysis. The results illustrated that the RF technique attains the reliable classification of battery physical properties and leads to the impressive quantification of both correlations and feature importance.

Recent advancements in deep learning techniques have considerably increased the capacity of complex data analysis.^{13,14} Besides, deep learning technology has been designed to overcome the requirement for prediction problems since it is particularly advantageous for extremely complicated nonlinear fittings of ANNs. New challenges for complex prediction problems, including accurate prediction of RUL battery, are expanded by the deep learning platform. Numerous innovative deep learning methods have provided plenty of advantages for battery health monitoring in recent years.^{15,16}

Numerous research activities concentrated on models based on the deep neural network (DNN), such as the ensemble learning, deep belief network, and extreme learning machine. These frameworks are often concentrated on fault measurement areas with less time series knowledge requirements.¹⁷ On the other hand, the convolution neural network and recurrent neural network (RNN) were proposed to the RUL field in recent years. For example, Li et al built a novel framework using compact convolutional neural network models through the concepts of transfer learning to improve the battery health estimation accuracy.¹⁸ Another disadvantage of these methods is that the chosen prediction model possesses more input parameters and need to have time consecutive.¹⁹ It is noteworthy that the typical RNN algorithm suffers from connecting the current input to the relevant information long before the current state. However, there is still a problem of poor prediction ability using the data-driven approaches for battery health prediction owing to the nonlinear structure of the LiBs.

Compared with other data-driven methods, the Gaussian process regression (GPR) method is a class of Bayesian model that has strong nonlinear modeling capability to solve and predict the regression problems.²⁰⁻²² Moreover, the GPR method can also improve prediction accuracy without the physical model. However, the trend fitting deteriorates when test data are far from the training data, and the predictive results are unsatisfactory. In this regard, some attractive papers have been published for calendar aging prediction of Li-ion batteries using the modified GPR method. For instance, a mechanism-conscious GPR model has been constructed for battery cycle life prediction. In this way, by coupling the polynomial equation and Arrhenius law into a compositional kernel through the GPR model, a modified model of the GPR was made, which could have an acceptable prediction against uncertainties.²³ In other interesting work, an

advanced Gaussian filter technique has been performed to obtain the smoothing incremental capacity curves. Then the health indexes (HIs) have been extracted from the partial incremental capacity curves as the input features of the GPR model. The results were demonstrated that the proposed model has advantages of high accuracy and robustness.²⁴

On the other hand, long short-term memory (LSTM)-RNN is another class of RNN that could solve some drawbacks of simple RNN, such as vanishing and exploding gradient.²⁵ Li et al implemented an architecture using LSTM-based time series processing, which allows the input charging curves to be variable in time steps and prediction to be attained even with incomplete sensor data.²⁵ Although, LSTM's configuration is complex and is composed of three gates, including the output gate, input gate, and forget gate.

However LSTM-based RNN models have attained state-of-the-art accomplishment on various machine learning tasks, the gating mechanism leads to significant complexity. As an alternative, the gated recurrent unit (GRU) architecture is similar to the LSTM architecture but has one fewer gate. Compared with an LSTM-based model, a GRU-based model due to the merging of the cell state and the hidden state has a more straightforward structure and fewer tensor operations (about 25% fewer), thus making model training easier and making it a very appropriate candidate for embedded implementations. So far, a few works have been done using GRU methodology in battery RUL prediction, but there are still weaknesses.^{26,27} For instance, Song et al²⁶ proposed this method to predict battery degradation. However, this study has not been considered the battery features and just applied capacity observation as input and has not been used multivariate time series prediction. In other work, Ungurean et al²⁷ proposed online state of health (SOH) estimation for LiBs using GRU. In the first step, they estimated the state of charge (SOC) and then used battery capacity to predict SOH. However, estimation of SOC and then using it to predict SOH is complex and will result in many errors. This method will be involved in two predictions that if the SOC estimation error is high, the SOH estimation error will be more. It is worth noting that they used univariate GRU for prediction. In this paper, a data-driven precise battery RUL prediction model is developed using an effective training network to cope with these limitations.

1.2 | Motivations and contributions

A GRU is a modern deep learning network to overcome the abovementioned problems. For verifying the

effects, the comparison of a number of state-of-the-art models with the suggested GRU in the present study is performed. Dissimilar to the above RNN-based approaches, and the GRU-RNN-based RUL approach is proposed to create nonlinear mapping among the battery capacity and observable variables. In particular, the GRU-RNN is an enhanced type of simple RNN to resolve a short-term dependence problem. Compared to the equivalent circuit models and electrochemical models that encompass differential equations, the GRU-RNN does not need to extract the battery's internal parameters and many other tasks to parametrization. The key contributions of the present work were summarized as follows:

1. Since time series data are time-dependent, there is a need to consider time delays (lag) to predict such cases that have not been addressed in most papers in battery degradation prediction. Toward this end, a deep learning method is introduced for multivariate time-series prediction.
2. A GRU model is proposed with multivariate input to predict the battery RUL. Unlike the LSTM, this method has fewer parameters due to merging the cell state and the hidden state and does not need a memory unit. Therefore, it makes a more straightforward structure and fast training.
3. Feature extraction through the statistical equations and feature selection is done based on RF. At this stage, not only the computational burden of modeling is reduced, but also the performance of the model is improved.
4. An adaptive learning rate optimization algorithm, namely Adam Optimization, is applied for the GRU-RNN model to optimize the training network. This technique cannot only complete the model training rapidly and stably but also reduce the effect of learning rate and the training time. Besides, an early stop technique is used to prevent overfitting.
5. The investigations on a reliable battery dataset from NASA demonstrate that GRU-RNN can obtain greater precision than the LSTM and traditional methods.

1.3 | Organization of the paper

The rest of the article is structured as the following: Section 2 outlines the history of the proposed method as well as its architecture, Section 3 discusses the new approach for RUL prediction related to model optimization and feature engineering in detail. Section 4 is about the results of RUL prediction and discussion, and finally, Section 5 summarizes the conclusions.

2 | RELATED WORK

2.1 | RNN architecture

RNN is a DNN class applied to assess time dependencies and input features on a sequential input to predict future output, with particular characteristics called internal cell state or memory. Therefore, each neuron's output varies depending on the current input and the background of previously hidden state outputs. Figure 1 indicates the function of the unfolded RNN structure with a feedback loop on a simple RNN, which can retain background information efficiently according to the number of time steps.²⁸

The assumption is that the input and output vectors of the RNN are $\mathbf{X}(t) = \{x_0, x_1, \dots, x_t\}$ and $\mathbf{Y}(t) = \{y_0, y_1, \dots, y_t\}$, respectively, which can have arbitrary dimensions. Hence, the battery dataset applied to train the model can be illustrated as the following:

$$\Psi = \{X, Y\} \tag{1}$$

Here, $x_t = [I, V, T]$ and $y_t = [C(t)]$, where

$$\begin{cases} I = [I_1, \dots, I_m], & m = \text{number of current - related features.} \\ V = [V_1, \dots, V_n], & n = \text{number of voltage - related features.} \\ T = [T_1, \dots, T_p], & p = \text{number of temperature - related features.} \end{cases}$$

and $C(t)$ is the measured battery capacity at cycle t .

RNNs operate by an iterative update of a hidden state, h , that is also a vector with arbitrary dimensions. First, at any step t , the next hidden state h_t is determined with the next input x_t and the hidden state h_{t-1} . Secondly, h_t is used to measure the next y_t output. The equations that mathematically define a single RNN cell in a single-layer RNN can be seen Equations (2) and (3) as follows:

$$h_t = \tanh(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \tag{2}$$

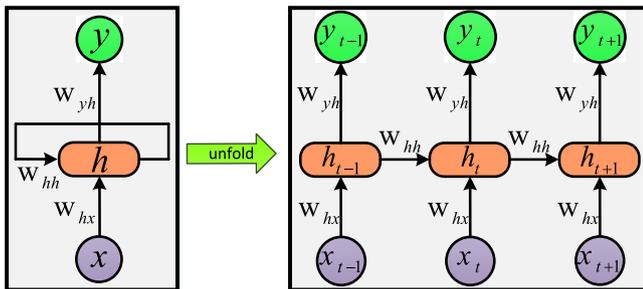


FIGURE 1 The structure of the simple RNN and unfolded RNN [Colour figure can be viewed at wileyonlinelibrary.com]

$$y_t(t) = W_{yh}h_t + b_y \tag{3}$$

where W_{hx} , W_{hh} , and W_{yh} imply the weights of each step. It should be noted that what makes the RNN recurrent is to apply the same weights in each step. In particular, only three sets of weights are used by a typical vanilla RNN to make calculations. W_{hh} is employed for all $h_{t-1} \rightarrow h_t$ links, W_{xh} is applied for all $x_t \rightarrow h_t$ links, and W_{yh} is employed for all $h_t \rightarrow y_t$ links. Also, a couple of biases have been applied for RNN: b_h and b_y .

2.2 | GRU-RNN architecture

The RNN may be used as a network memory, different from the feedforward neural network. Consequently, the current state is associated with the previous state and the current input. That helps the RNN handle time series problems by storing, preserving, and evaluating the previous complex signals over a specific time. RNNs are extensively applied in numerous applications such as prediction of time series, system modeling, and natural language processing. Nevertheless, complex hidden layers and long time series may contribute to the exploding and vanishing of gradients throughout back-propagation procedures. Across all the enhanced RNNs, the GRU-RNN not only has a simple structure but also able to capture long-term sequential dependencies. Furthermore, the gradients are more resistant to vanishing compared to other RNNs, and fewer memory resources are needed. Therefore, GRU-RNN is ideal for dealing with highly correlated issues with time series, such as RUL prediction of the battery system. The structure of the GRU-RNN cell and its deep learning model is shown in Figures 2 and 3, respectively, which will be explained in Section 3.2.

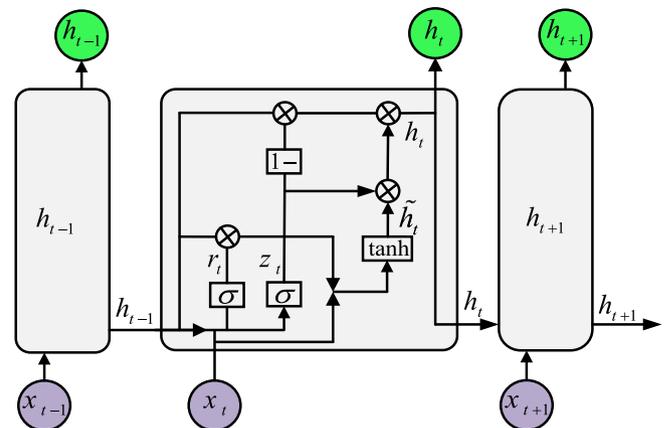


FIGURE 2 The structure of gated recurrent unit memory cell [Colour figure can be viewed at wileyonlinelibrary.com]

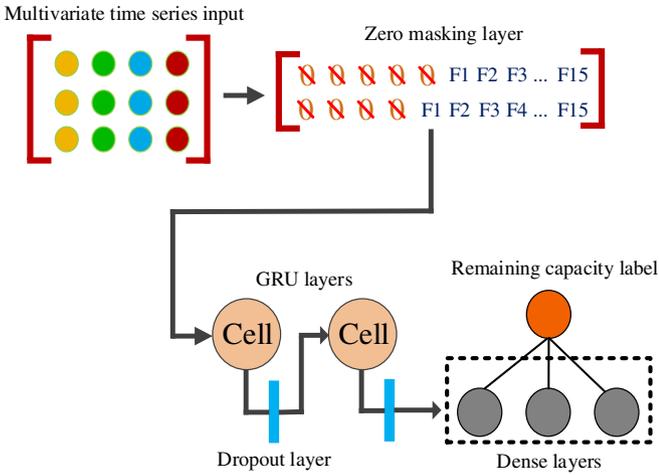


FIGURE 3 The structure of the deep learning model for RUL prediction [Colour figure can be viewed at wileyonlinelibrary.com]

Behind the GRU-RNN idea, there are two main key parameters, which are called update gate and reset gate. Both the reset gate r_t and update gate z_t are relevant to x_t and h_{t-1} . x_t is the corresponding input sequence, and h_{t-1} is the memory cell output at the previous time point. These two gates have distinct network functions. Four primary Equations (4)-(7) are used to calculate the GRU-RNN forward propagation.

- **Update gate:** The update gate is developed to monitor the previous data's effect on the current state. The bigger the updated value is, the more previous information is utilized to specify the current state. Equation (4) reflects the GRU-RNN update operation.

$$z(t) = \sigma(w_z \cdot [h_{t-1}, x_t] + b_z) \quad (4)$$

- **Reset gate:** The reset gate controls the level of ignorance of information in h_{t-1} . If the value of the reset gate is small, the information is more overlooked. This parameter can be applied for the prediction of RUL of LIBs for rejecting outliers, noises, and unnecessary degradation information between adjacent cycles.²⁹ This is because the structure of the RNN-based model considers long-term information, and which leads to suppress the effect of weight at the adjacent cycle data.³⁰ Equations (5) and (6) reflect the GRU-RNN reset operation.

$$r(t) = \sigma(w_r \cdot [h_{t-1}, x_t] + b_r) \quad (5)$$

$$\tilde{h}(t) = \tanh(w_{\tilde{h}} \cdot [r_t \odot h_{t-1}, x_t] + b_{\tilde{h}}) \quad (6)$$

- **Output:** Equation (7) indicates the GRU-RNN output operation.

$$h(t) = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (7)$$

The derivatives of σ and \tanh are the function of the primary function, therefore the derivatives can be computed by the primary functions.

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (8)$$

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad (9)$$

3 | GRU-RNN-ORIENTED RUL PREDICTION

To clarify the steps taken for both feature engineering and RUL prediction methods, the framework of the proposed method is indicated in Figure 4.

3.1 | Feature engineering

Before training the model, it is necessary to do pre-processing on raw data. The statistical characteristics are effective methods with the advantages of low time consumption for calculation and also simplicity of implementation.³¹ Many studies have been used statistical features for diagnosis and prognosis problems,³² and it has shown strong results. RMS, Kurtosis, Skewness, Peak-Peak, Mean are the most commonly used time-domain features in industrial applications.³³

The measured terminal voltage and load current of any battery will vary as it is charged and discharged. The essential characteristics such as the nominal voltage of the cell, peak charged, and end of life (EOL) can be extracted from each charge and discharge curve in cycles. In this work, battery characteristic during discharge mode has been used for feature extraction. The terminal and current-voltage profiles of a charge-discharge life cycle for 4 battery cells (called B0005, B0006, B0007, and B0018) are indicated in Figure 5 which will be explained in detail in Section 4.1.

In our case, the feature engineering problem is divided into three parts:

1. 30 time-domain features are extracted from battery signals, including voltage, current, and temperature, based on statistical equations. These formulas are listed in Table 1.

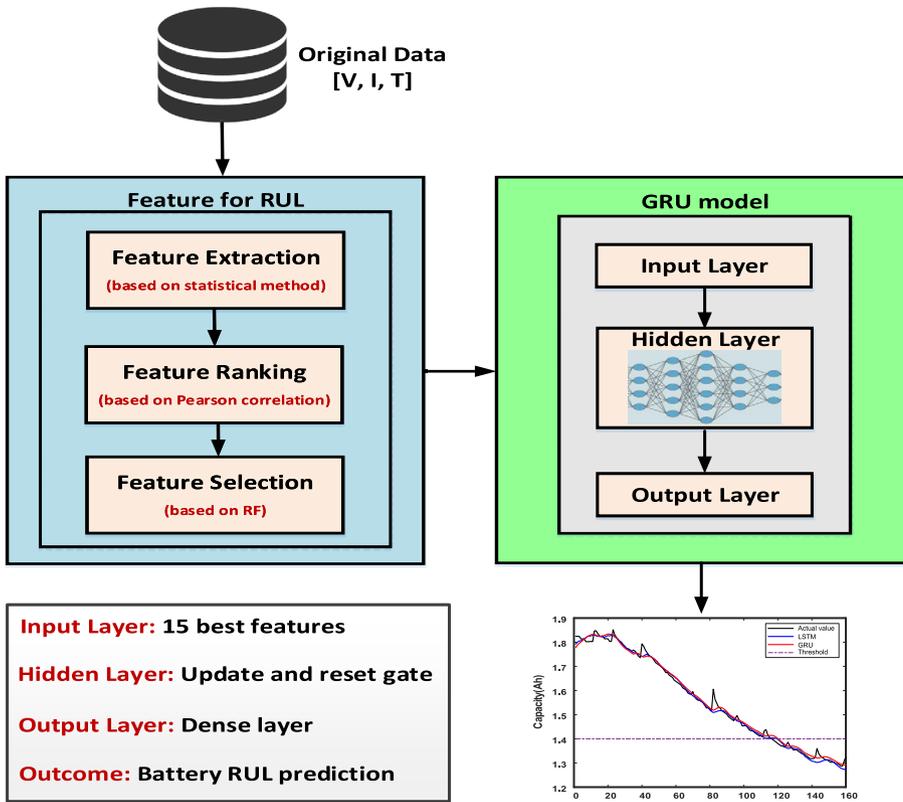


FIGURE 4 Framework of battery remaining useful life prediction [Colour figure can be viewed at wileyonlinelibrary.com]

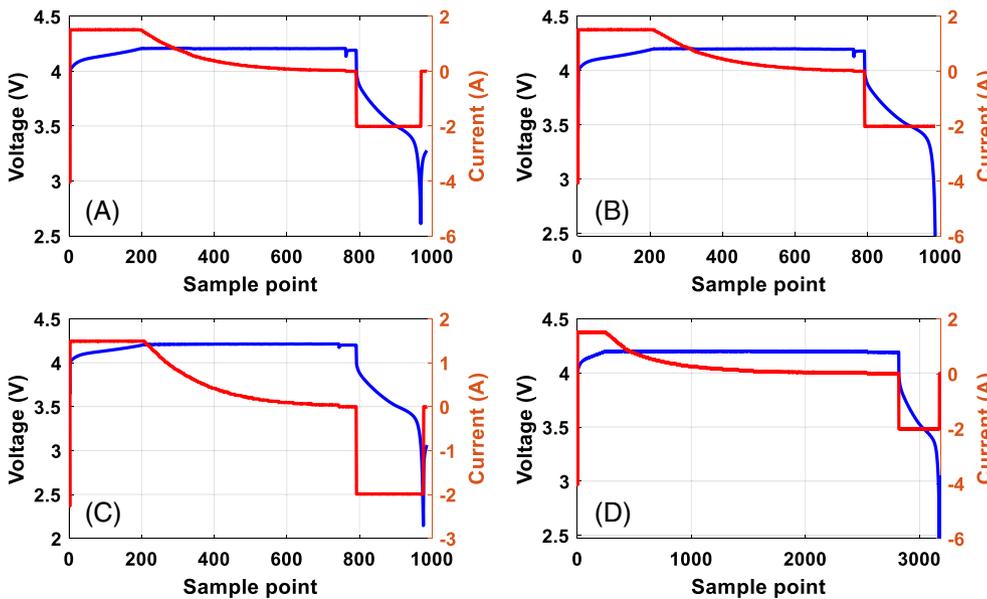


FIGURE 5 The terminal and current-voltage profiles of a charge-discharge life cycle; A, B0005, B, B0006, C, B0007, and D, B0018 [Colour figure can be viewed at wileyonlinelibrary.com]

2. These features are sorted according to feature importance and separated the features based on the Pearson correlation, which has correlated with the coefficient above 0.5. For the purpose of quantitative confirmation of the linear correlation between the capacity and extracted features, Pearson correlation analysis can be used, which is computed as Reference 33:

$$r = \frac{\sum_{i=1}^n (HI_i - \bar{HI})(C_i - \bar{C})}{\sqrt{\sum_{i=1}^n (HI_i - \bar{HI})^2} \sqrt{\sum_{i=1}^n (C_i - \bar{C})^2}} \quad (10)$$

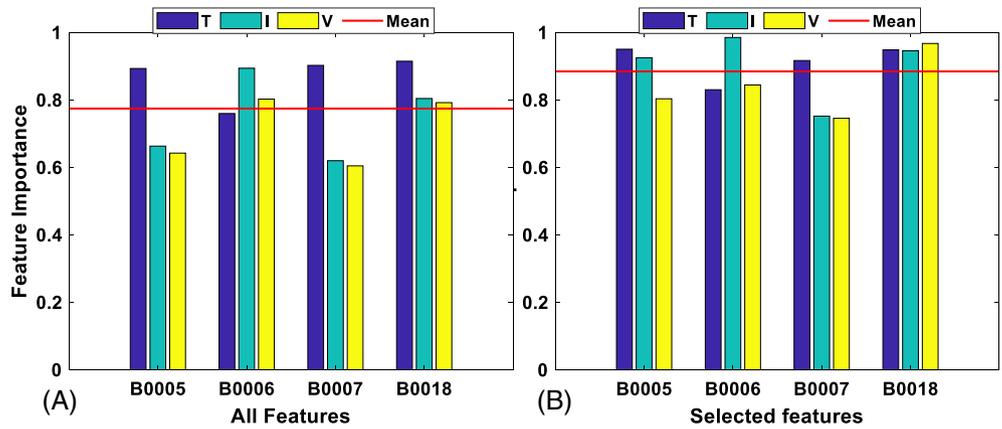
where HI , C , \bar{HI} , and \bar{C} denote to feature, capacity, mean values of the HI , and mean values of the capacity,

TABLE 1 Statistical formulas in the time-domain

Name	Formula	Name	Formula	Name	Formula
Mean	$F_m = \frac{1}{N} \sum_i^N x(i)$	Shape factor (SHF)	$F_{shf} = \frac{F_{rms}}{\frac{1}{N} \sum_i^N x(i) }$	Skewness factor (SF)	$F_{sf} = \frac{\frac{1}{N} \sum_i^N x(i) ^3}{F_{rms}^3}$
SD (STD)	$F_{std} = \sqrt{\frac{1}{N} \sum_i^N (x(i) - \bar{x})^2}$	Crest factor (CF)	$F_{cf} = \frac{F_p}{F_{rms}}$	Kurtosis factor (KF)	$F_{kf} = \frac{\frac{1}{N} \sum_i^N x(i) ^4}{F_{rms}^4}$
Root mean square (RMS)	$F_{rms} = \sqrt{\frac{1}{N} \sum_i^N x(i)^2}$	Impulse factor (IF)	$F_{if} = \frac{F_p}{\frac{1}{N} \sum_i^N x(i) }$	Clearance factor (CF)	$F_{clf} = \frac{F_p}{\frac{1}{N} \sum_i^N \sqrt{ x(i) }}$
Peak	$F_p = \max x(i) $	—	—	—	—

Note: $x(i)$ is the battery signals series; \bar{x} is the mean value of the series.

FIGURE 6 Feature extraction and selection: A, all statistical features coefficient correlation and B, selected features correlation coefficient [Colour figure can be viewed at wileyonlinelibrary.com]



respectively. The value of the Pearson correlation coefficient r ranges between -1 and $+1$. If the correlation is equal to ± 1 , features and capacity are completely correlated linearly, and no correlation exists in the case that it equals 0.

3. Finally, feature selection has been made based on the RF algorithm owing to eliminating the less-relevant features and effective data training. To illustrate the importance of pre-processing, Figure 6 shows that the average of the extracted features coefficient for pre-processing is 0.76, and after pre-processing is 0.88. And also Figure 7 indicates the heatmap of selected features using RF algorithm. To better understand the problem, the process of selecting the best features by RF algorithm is described below.

RF is an intelligent ensemble learning algorithm based on a decision tree, which contains a group of structured tree classifiers $h(x, \Theta_k)$, ($k = 1, 2, 3, \dots$), in which a unit vote is cast by each tree for the especially known class at input x and Θ_k is identically distributed random vectors and also independent.³⁴ A margin

function $mg(\cdot)$, which is referred to as the confidence level for the RF model, needs to be defined with a performance index,

$$mg(x, y) = av_k I(h_k(x, \Theta_k) = y) - \max_{j \neq y} av_k I(h_k(x, \Theta_k) = j) \tag{11}$$

In this equation, $I(\cdot)$ implies the indicator function, as well as $av(\cdot)$, is a mean value. This index is divided into two terms: the first term is referred to as the average number of votes at (x, y) for the right class, and the other term implies the average vote for the most class excluding the right class. In the case of large values of the margin, the confidence level will be high. Afterward, the generalization error PE^* is obtained through the following equation:

$$PE^* = P_{x,y}(mg(\mathbf{x}, \mathbf{y}) < 0) \tag{12}$$

where $P(\cdot)$ denotes the probability. By increasing the number of trees, the convergence of nearly all sequences Θ_k , PE^* is tended toward the following equation:

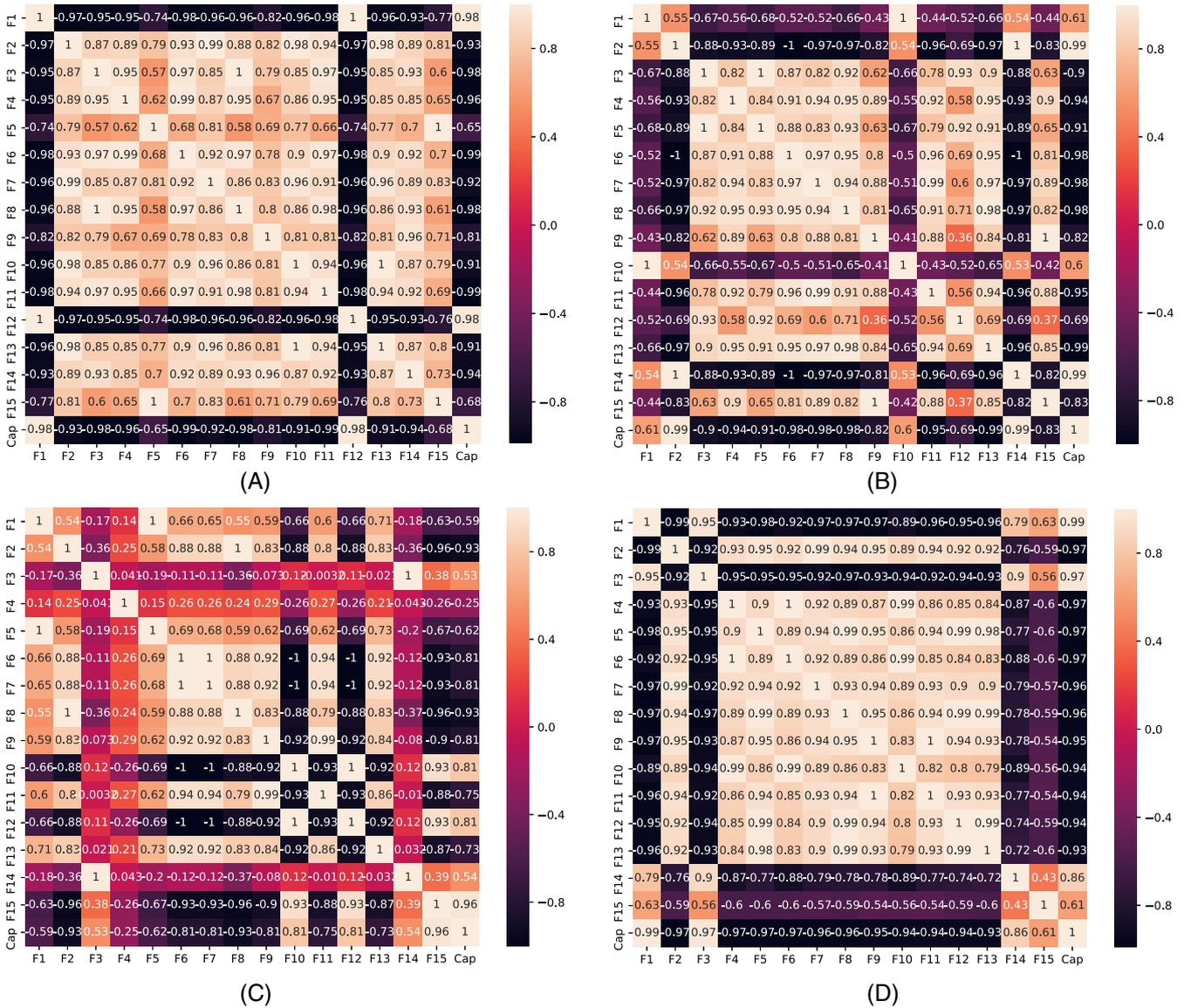


FIGURE 7 Heatmap of selected features using RF algorithm; A, B0005, B, B0006, C, B0007, and D, B0018 [Colour figure can be viewed at wileyonlinelibrary.com]

$$P_{x,y}(P_{\Theta}(h(\mathbf{x}, \Theta) = \mathbf{y}) - \max_{j \neq y} P_{\Theta}(h(\mathbf{x}, \Theta) = j) < 0) \quad (13)$$

When the generalization error is converged, a considerably low value for generalization error can be produced by RF. Also, RF does not overfit by adding more trees. The upper bound for the PE^* is determined by

$$PE^* \leq \frac{\rho(1-s^2)}{s^2} \quad (14)$$

where ρ denotes the correlation average value, s implies the strength of each tree in the RF model. Hence, by decreasing the correlation among trees and raising the strength of each tree, the RF model would give higher

precision of the predictions. For the purpose of showing the performance of the features selected by RF, the heatmap is plotted which F_1, F_2, \dots, F_{15} , are 15 selected features and has a high correlation with Cap as output, which is the battery capacity.

3.2 | GRU-RNN training

As it can be seen in Figure 3, one input layer fed into a GRU layer with 50 neurons was applied to build the GRU-RNN. In turn, this layer is fed into two hidden layers with 50 neurons for both of them, which then fed into a fully connected dense layer of 20 neurons. Given

that the battery data is time series, this work takes into account the time dependency, which has been overlooked in most existing papers, and it was not a delay (time lag) for training data. Due to the use of multivariate time series input, the zero masking layer has been considered for sequence processing. Moreover, the third layer is a fully connected dense layer applying a linear transformation for achieving the RUL prediction results using the sigmoid activation function, which is performed as follows:

$$RUL_t^* = \sigma(W_t \cdot h_t + b_s) \quad (15)$$

where W_t and b_s respectively denote the weight vector and biases of the fully connected layer at time step t . The mean absolute error (MAE) is chosen as the loss function, and it is calculated as indicated in Equation (16):

$$L = \left(\sum_{t=1}^l (RUL_t - RUL_t^*)^2 \right) / l \quad (16)$$

where RUL_t implies the measured value, RUL_t^* denotes the predicted value, and l is the length of the battery discharge cycles.

3.3 | Adam optimization algorithm

Deep learning scientists have often sought to improve the model's efficiency and loss function value by the model's training epochs. Stochastic gradient descent (SGD) is one of these approaches that give a single learning rate for all weight updates and does not modify the learning rate throughout the training.³⁵ Nevertheless, this method is not effective for the training model due to frequent fluctuations; it will keep overshooting near to the desired exact minima and very time-consuming to converge to the correct network weights, which is inapplicable for online battery RUL prediction. Root mean square propagation (RMSprop) is another commonly used optimization method that overcomes the decaying learning rate problem of the SGD method. However, both of them still have the problem of different momentums for different parameters.

Therefore, the Adam algorithm was proposed by Kingma and Ba³⁶ to introduce the concept of adaptive momentum along with the adaptive learning rate, which computes the exponentially decaying average of previous gradients along with an adaptive learning rate. The key advantages of using the Adam algorithm in convex optimization issues are invariant to limited memory

requirements and diagonal rescale of gradients. The Adam optimizer is a hybrid version of the AdaGrad, and RMSProp algorithms.^{37,38}

3.4 | Early stopping technique to prevent overfitting

Early stopping is widely used to implicitly regularize some convex learning problems.³⁹ Since the understanding and implementation are simple and have been reported to be superior to regularization methods in many studies, for example.⁴⁰ During the training, the model is evaluated on a holdout validation dataset after each epoch. The training process is stopped if the model's performance on the validation datasets starts to deteriorate (ie, the loss is beginning to rise or accuracy is beginning to decrease). This technique is referred to as an early exit, so this is called an early stopping and is one of the most frequently used ways of regularizing neural networks. Its success lies in its quality and simplicity. In some papers, results confirm that early stopping could potentially improve generalization performance.

4 | RESULTS AND DISCUSSION

Matlab 2019 performed data preprocessing and feature engineering in this work, and Python 3.6 was also used for model simulation and training. The simulation was performed on a laptop with a graphic card NVIDIA GeForce 930 M at 6 GB, 64-bit operating system and an Intel Core i7 – 6500U processor (6 MB cache, up to 3.18 GHz), x64-based processor.

For a fair comparison, model parameters (such as learning rate = $1e-5$, lag = 8, epoch = 1000) are considered the same values. Besides, the batch size tuning has been done by a common method, which is called the grid search method. We set an early stopping for both models if the validation accuracy does not increase for 1000 global steps; the training will stop. It is worth noting that, due to the use of the *reduceLR* technique, the impact of the learning rate has diminished, and it has been no need for exact tuning.

4.1 | Data description and evaluation criteria

In the present article, the public battery dataset of NASA Ames Prognostics Center of Excellence is used to validate

our proposed method. Three different operational profiles (an impedance, discharge, and charge) were used at room temperature to run four Li-ion batteries (B0005, B0006, B0007, and B0018). In this article, the authors only used the discharging mode for RUL prediction. Discharging was performed at a constant current level of 2A until the falling of the voltage of batteries B0005, B0006, and B0007 to 2.7, 2.5, and 2.2 V, respectively. Repeated charge and discharge cycles lead to rapid battery aging, although impedance measurements give an overview of internal battery parameters altering with the progression of the aging process. The tests were completed in the case that the battery reached EOL criteria on a 30% fade in rated capacity (from 2 to 1.4 Ahr).⁴¹

The concept of building a validation set is to evaluate the model's performance prior to applying it to make predictions. The development of a validation set for time series issues is challenging since the time component must be considered. They represent calculation precision and are often applied to compare the pros and cons of algorithms. As the train-test-split or k-fold validation cannot be used directly, the pattern will be disrupted in the series. Hence, three evaluation criteria of RUL prediction error, root mean square error (RMSE), and MAE are used to calculate and demonstrate the suggested approach's precision and stability. The equations are given in Equations (4)-(19).

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (\hat{Y}(i) - Y(i))^2} \quad (17)$$

$$MAE = \frac{1}{m} \sum_{i=1}^m |(\hat{Y}(i) - Y(i))| \quad (18)$$

$$RUL_{error} = RUL_{predict} - RUL_{true} \quad (19)$$

where $Y(i)$ and $\hat{Y}(i)$ denoted to the predicted capacity and measurement capacity series, respectively. i is the number of cycles between the actual battery and the first prediction cycle. Besides, to assess the uncertainty quantification of the proposed model, the 95% confidence interval (CI) is performed for evaluation of the uncertainty as

$$95\%CI = \hat{Y}(i) \pm 1.96 \times \sigma^2(\hat{Y}(i)) \quad (20)$$

where 95%CI is the confidence interval for RUL prediction. $\hat{Y}(i)$ and σ^2 denote mean values of RUL prediction and variance of the predicted values, respectively.

4.2 | RUL prediction results for battery degradation data

Four datasets from Figure 8 show an accelerating aging process obtained from the discharge mode of the battery. It shows that the battery degradation goes down during the time due to internal reactions in charging and discharging cycles. To verify the proposed model, we considered two scenarios, including training the model with 60% dataset and the other one with 80%. In the first scenario, Figure 9 shows the battery RUL prediction with 60% training data for four different cases in which the start point for prediction is from the 97th cycle for B0005, B0006, B0007, and 75th cycle for B0018. Figure 10 shows the prediction error for both GRU-RNN and LSTM-RNN as well. To indicate the GRU-RNN model's accuracy, a comparison has been made between the GRU-RNN, LSTM-RNN, and support vector machine (SVM) methods. Figure 11 shows a box plot of all training and testing errors together. GRU-RNN has the lowest error for all the cases, and the LSTM-RNN has less accuracy than GRU-RNN. Meanwhile, the SVM has a big difference from them, which is not suitable for the long-term dependency prediction. Given that the evaluation of models with MAE, RMSE, and RUL prediction, Table 2 provided a comparison between the models. For the second scenario, we applied 80% data for training and compared the different methods. Figure 12 shows the battery RUL prediction with 80% training data for four different cases, which the start point for prediction is from the 129th cycle for B0005, B0006, B0007, and 100th cycle for B0018. The prediction error is shown in Figure 13, and also the box plot of all training and testing errors in Figure 14 illustrates the GRU-RNN has less amplitude of error than the other methods. For better methods

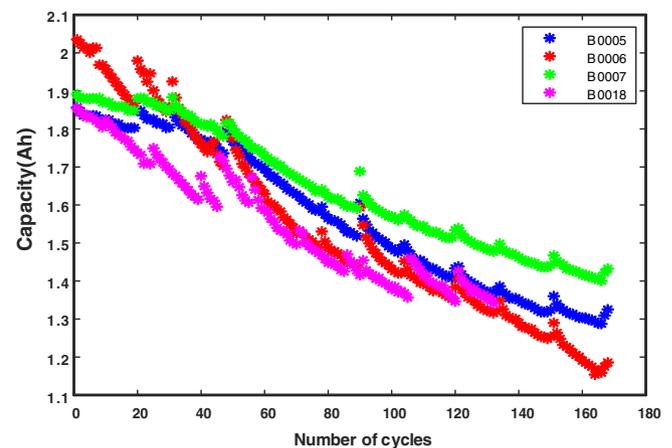


FIGURE 8 NASA dataset [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 9 RUL prediction (60% of training): A, B0005, B, B0006, C, B0007, and D, B0018 [Colour figure can be viewed at wileyonlinelibrary.com]

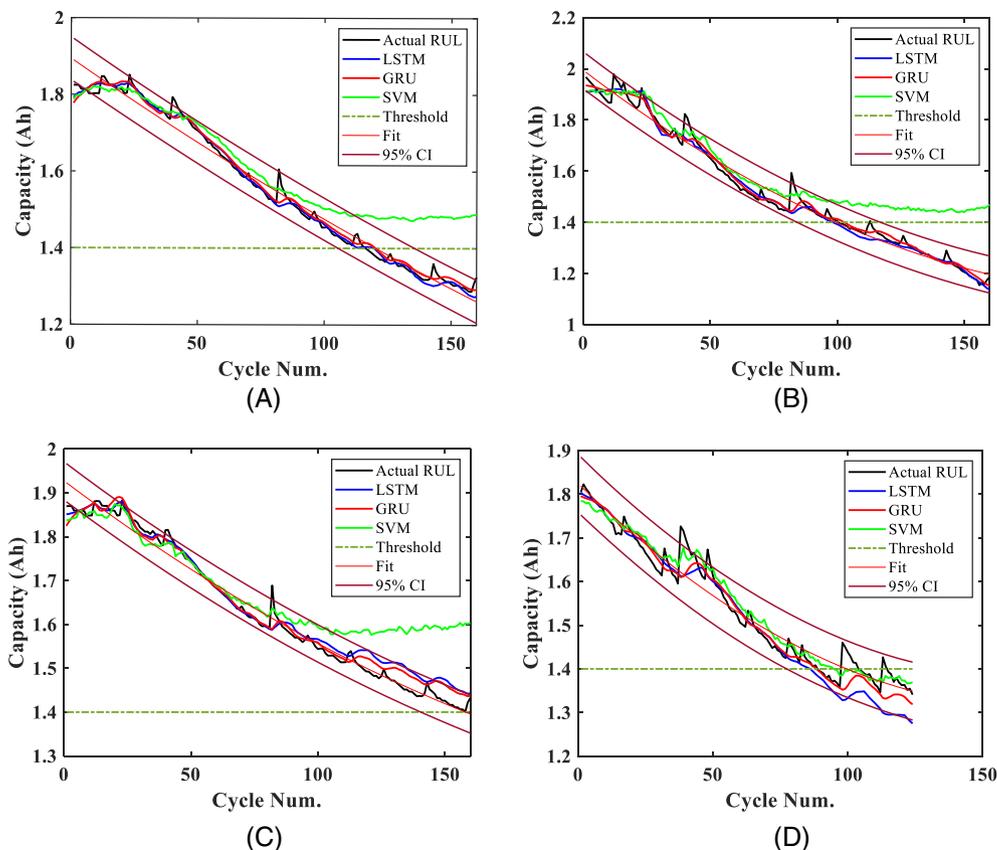
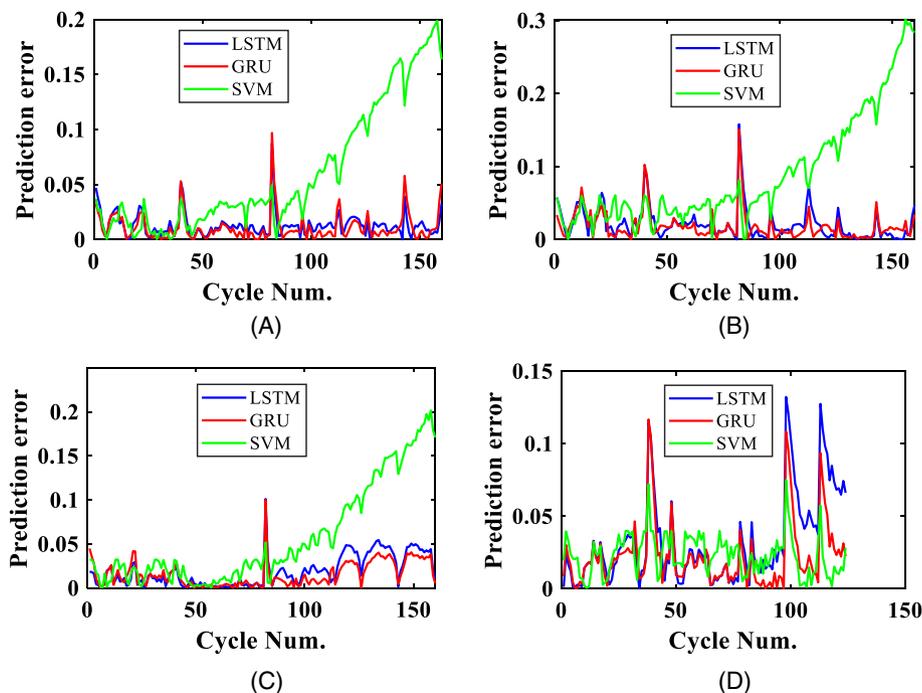


FIGURE 10 Prediction error (60% of training): A, B0005, B, B0006, C, B0007, and D, B0018 [Colour figure can be viewed at wileyonlinelibrary.com]



comparison, Table 3 is provided the MAE, RMSE, and RUL prediction error. The results achieved in the above section show that the deep learning models are much

more precise compared to a conventional method because of advantages such as capturing long-term dependency time series data.

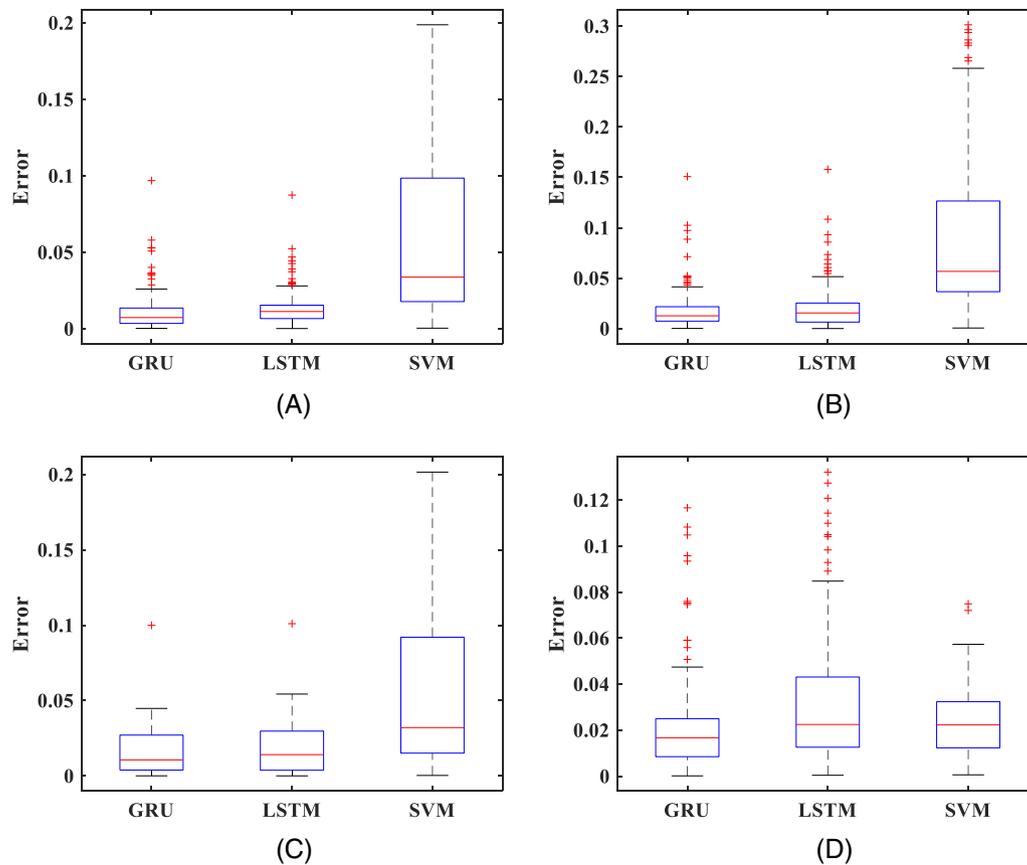


FIGURE 11 Box plot of prediction error (60% of training): A, B0005, B, B0006, C, B0007, and D, B0018 [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 RUL prediction results with 60% of training

Battery cell	Method	MAE	RMSE	RUL error
B0005	GRU	0.0115	0.0145	0.6213
	LSTM	0.0124	0.0174	0.6337
	SVM	0.1328	0.1385	6.7772
B0006	GRU	0.0127	0.0165	0.6510
	LSTM	0.0136	0.0211	0.6878
	SVM	0.1777	0.1884	9.0664
B0007	GRU	0.0268	0.0290	1.3680
	LSTM	0.0367	0.0392	1.8753
	SVM	0.1283	0.1349	6.5481
B0018	GRU	0.0267	0.0389	1.0681
	LSTM	0.0561	0.0657	2.2460
	SVM	0.0302	0.0455	1.8116

In addition, to highlight the proposed method, the average RMSE percentage for all cases has been listed in Table 4. This table shows the GRU-RNN error is about 2% which is more accurate compared to its peer and

appropriate for real-world systems. Moreover, the executed time for GRU-RNN is about 14 seconds which demonstrates that due to fewer parameters, the learning speed is faster than the speed of LSTM (19 seconds). However, SVM has a high execution speed due to its straightforward structure, but it is not suitable for time series problems and has very weak prediction accuracy.

5 | CONCLUSION

As a crucial tool for PHM, the RUL prediction is capable of ensuring a possible Li-ion battery failure time in advance. One of the most crucial concerns in the RUL prediction of Li-ion batteries is the way of appropriately learning the long-term dependencies of several hundred cycles while limited degradation data are available.

This paper has been presented for a data-driven model to monitor battery health. The GRU RNN has been used to predict the battery RUL. To achieve high accuracy prediction, important features based on Pearson correlation and RF algorithm have been applied to feed

FIGURE 12 RUL prediction (80% of training); A, B0005, B, B0006, C, B0007, and D, B0018 [Colour figure can be viewed at wileyonlinelibrary.com]

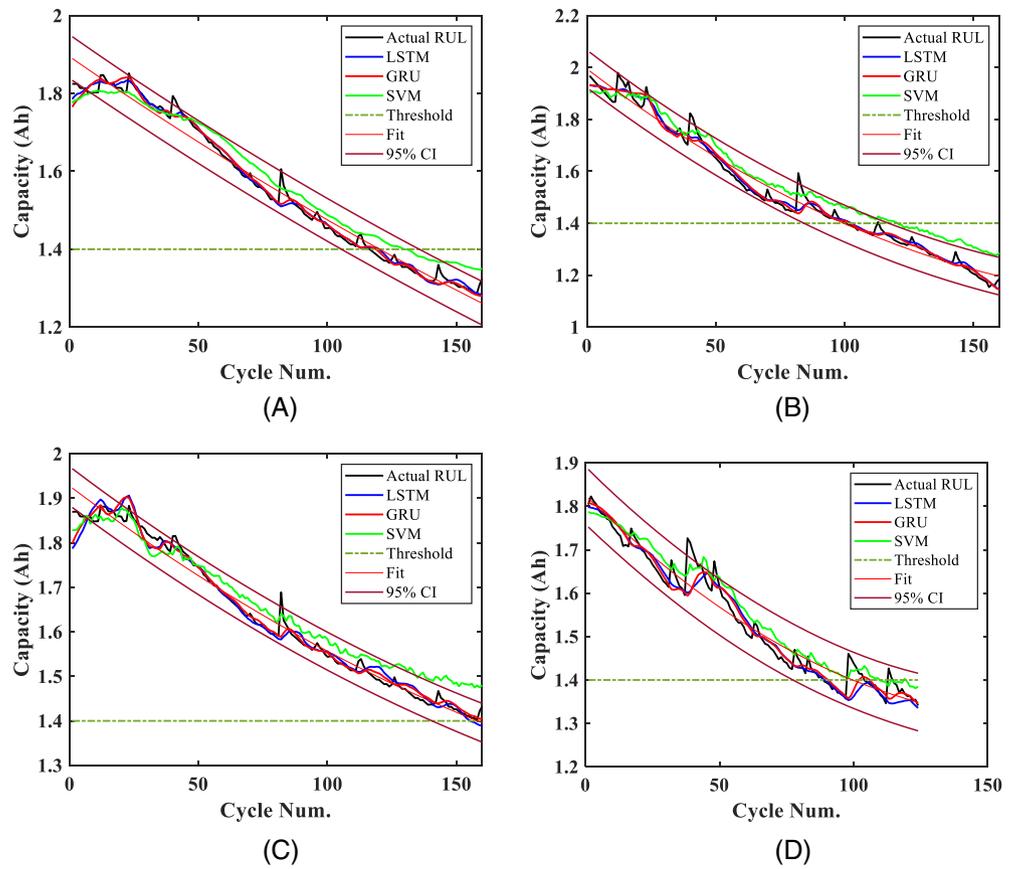
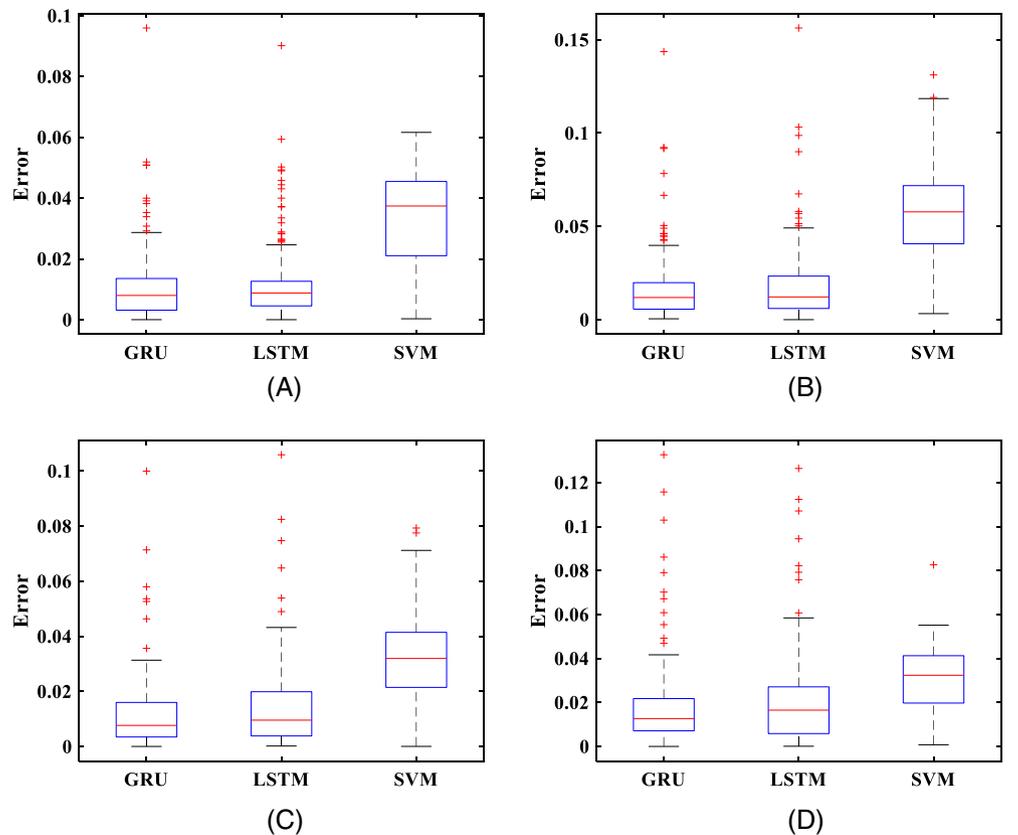


FIGURE 13 Box plot of prediction error (80% of training); A, B0005, B, B0006, C, B0007, and D, B0018 [Colour figure can be viewed at wileyonlinelibrary.com]



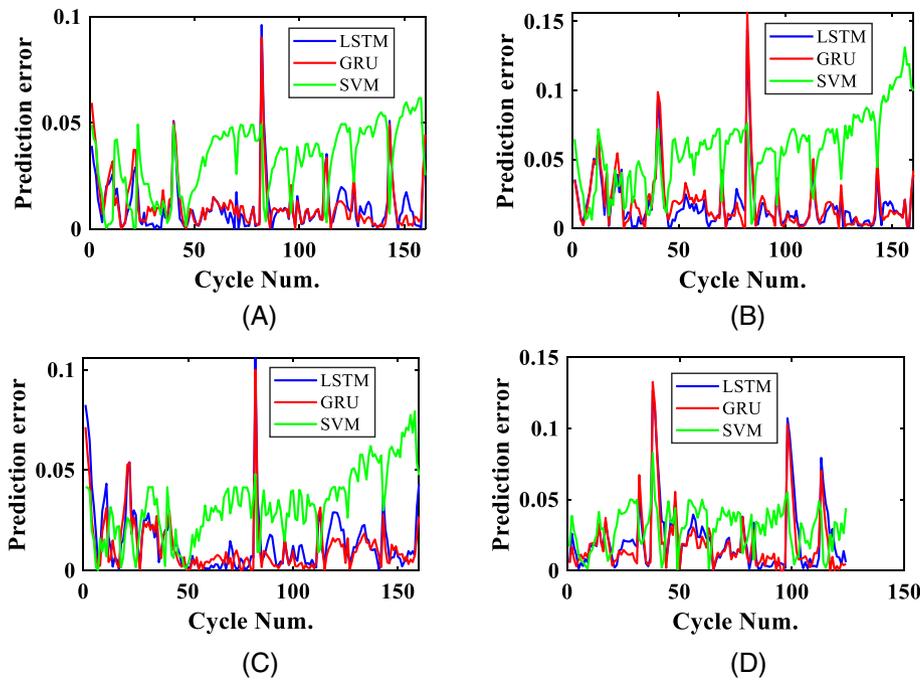


FIGURE 14 Prediction error (80% of training); A, B0005, B, B0006, C, B0007, and D, B0018 [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 3 RUL prediction results with 80% of training

Battery cell	Method	MAE	RMSE	RUL error
B0005	GRU	0.0099	0.0156	0.1698
	LSTM	0.0117	0.0158	0.1990
	SVM	0.0500	0.5122	0.8504
B0006	GRU	0.0133	0.0165	0.2267
	LSTM	0.0147	0.0172	0.2504
	SVM	0.1042	0.1050	1.7744
B0007	GRU	0.0089	0.0106	0.1526
	LSTM	0.0115	0.0155	0.1958
	SVM	0.0615	0.0624	1.0461
B0018	GRU	0.0163	0.0259	0.2295
	LSTM	0.0249	0.0337	0.3492
	SVM	0.0265	0.3025	0.3722

TABLE 4 The average of RMSE prediction accuracy and executed time for all cases

	SVM	LSTM	GRU
RMSE (%)	17.43	2.82	2.09
Executed time (s)	0.9511	19.2397	14.1173

into the GRU-RNN as a multivariate input. Moreover, to optimize the training network, the Adam technique has been applied for convex optimization, which requires low memory. At the same time, an early stopping technique

has been used to deal with overfitting and leads to enhance the performance of the GRU-RNN model.

For the experimental and evaluation of our proposed method, the NASA Li-ion battery dataset has been applied. The findings have been compared with its sibling technique, which is called LSTM. The results highlight the proposed method has higher accuracy and efficiency than LSTM-RNN and SVM.

ORCID

Reza Rouhi Ardehshiri <https://orcid.org/0000-0002-6619-929X>

Chengbin Ma <https://orcid.org/0000-0002-0221-8084>

REFERENCES

- Liu K, Li K, Peng Q, Zhang C. A brief review on key technologies in the battery management system of electric vehicles. *Front. Mech. Eng.* 2019;14(1):47-64.
- Liu Z, Mohammadzadeh A, Turabieh H, Mafarja M, Band SS, Mosavi A. A new online learned interval type-3 fuzzy control system for solar energy management systems. *IEEE Access.* 2021;9:10498-10508.
- Naha A, Khandelwal A, Hariharan KS, Kaushik A, Yadu A, Kolake SM. On-board short circuit detection of Li-ion batteries undergoing fixed charging profile as in smartphone applications. *IEEE Trans. Ind. Electron.* 2019;66(11):8782-8791.
- El Mejdoubi A, Chaoui H, Gualous H, Van Den Bossche P, Omar N, Van Mierlo J. Lithium-ion batteries health prognosis considering aging conditions. *IEEE Trans. Power Electron.* 2018;34(7):6834-6844.
- Liao L, Kottig F. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an

- application to battery life prediction. *IEEE Trans Reliab.* 2014; 63(1):191-207.
6. Shen D, Wu L, Kang G, Guan Y, Peng Z. A novel online method for predicting the remaining useful life of lithium-ion batteries considering random variable discharge current. *Energy.* 2021;218:119490.
 7. Chen L, Wang H, Chen J, et al. A novel remaining useful life prediction framework for lithium-ion battery using grey model and particle filtering. *Int J Energy Res.* 2020;44(9):7435-7449.
 8. Chang Y, Fang H, Zhang Y. A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery. *Appl Energy.* 2017;206:1564-1578.
 9. Lyu Z, Gao R. Li-ion battery state of health estimation through Gaussian process regression with Thevenin model. *Int J Energy Res.* 2020;44(13):10262-10281.
 10. Shamsirband S, Fathi M, Chronopoulos AT, Montieri A, Palumbo F, Pescapè A. Computational intelligence intrusion detection techniques in mobile cloud computing environments: review, taxonomy, and open research issues. *J Information Security Appl.* 2020;55:102582.
 11. Shamsirband S, Rabczuk T, Chau K-W. A survey of deep learning techniques: application in wind and solar energy resources. *IEEE Access.* 2019;7:164650.
 12. Liu K, Hu X, Zhou H, Tong L, Widanalage D, Marco J. Feature analyses and modelling of lithium-ion batteries manufacturing based on random forest classification. *IEEE/ASME Trans Mechatronics.* 2021;1-1. <http://dx.doi.org/10.1109/tmech.2020.3049046>.
 13. Qummar S, Khan FG, Shah S, et al. A deep learning ensemble approach for diabetic retinopathy detection. *IEEE Access.* 2019; 7:150530.
 14. Shamsirband S, Fathi M, Dehzangi A, Chronopoulos AT, Alinejad-Rokny H. A review on deep learning approaches in healthcare systems: taxonomies, challenges, and open issues. *J Biomed Inform.* 2020;113:103627
 15. Hu X, Che Y, Lin X, Onori S. Battery health prediction using fusion-based feature selection and machine learning. *IEEE Trans Transp Electr.* 2020;7(2):382-398.
 16. Ardehshiri RR, Balagopal B, Alsabbagh A, Ma C, Chow M-Y. Machine learning approaches in battery management systems: state of the art: remaining useful life and fault detection. Paper presented at: 2020 2nd IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESSES); September 1-3, 2020; Cagliari, Italy:61-66:IEEE.
 17. Mahmoudi MR, Baleanu D, Qasem SN, Mosavi A, Band SS. Fuzzy clustering to classify several time series models with fractional Brownian motion errors. *Alex Eng J.* 2021;60(1):1137-1145.
 18. Li Y, Li K, Liu X, Wang Y, Zhang L. Lithium-ion battery capacity estimation—A pruned convolutional neural network approach assisted with transfer learning. *Appl Energy.* 2021;285:116410.
 19. Wang Z-H, Hendrick, Horng G-J, Wu H-T, Jong G-J. A prediction method for voltage and lifetime of lead-acid battery by using machine learning. *Energy Explor Exploit.* 2020;38(1):310-329.
 20. Liu K, Ashwin TR, Hu X, Lucu M, Widanage WD. An evaluation study of different modelling techniques for calendar ageing prediction of lithium-ion batteries. *Renewable Sustainable Energy Rev.* 2020;131:110017.
 21. Liu K, Li Y, Hu X, Lucu M, Widanage WD. Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries. *IEEE Trans Ind Infor.* 2019;16(6):3767-3777.
 22. Li X, Wang Z, Yan J. Prognostic health condition for lithium battery using the partial incremental capacity and Gaussian process regression. *J Power Sources.* 2019;421:56-67.
 23. Liu K, Hu X, Wei Z, Li Y, Jiang Y. Modified Gaussian process regression models for cyclic capacity prediction of lithium-ion batteries. *IEEE Trans Transp Electr.* 2019;5(4):1225-1236.
 24. Li X, Yuan C, Li X, Wang Z. State of health estimation for Li-ion battery using incremental capacity analysis and Gaussian process regression. *Energy.* 2020;190:116467.
 25. Li W, Sengupta N, Dechent P, Howey D, Annaswamy A, Sauer DU. Online capacity estimation of lithium-ion batteries with deep long short-term memory networks. *J Power Sources.* 2021;482:228863.
 26. Song Y, Li L, Peng Y, Liu D. Lithium-ion battery remaining useful life prediction based on GRU-RNN. Paper presented at: 2018 12th International Conference on Reliability, Maintainability, and Safety (ICRMS); October 17-19, 2018; Shanghai, China:317-322:IEEE.
 27. Ungurean L, Micea MV, Cârstoiu G. Online state of health prediction method for lithium-ion batteries, based on gated recurrent unit neural networks. *Int J Energy Res.* 2020;44(8):6767-6777.
 28. Venugopal P, Vigneswaran T. State-of-health estimation of Li-ion batteries in electric vehicle using IndRNN under variable load condition. *Energies.* 2019;12(22):4338.
 29. Zhang Y, Xiong R, He H, Pecht MG. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Trans. Veh Technol.* 2018;67(7): 5695-5705.
 30. Choi Y, Ryu S, Park K, Kim H. Machine learning-based lithium-ion battery capacity estimation exploiting multi-channel charging profiles. *IEEE Access.* 2019;7:75143-75152.
 31. Feng J, Kvam P, Tang Y. Remaining useful lifetime prediction based on the damage-marker bivariate degradation model: a case study on lithium-ion batteries used in electric vehicles. *Eng. Failure Anal.* 2016;70:323-342.
 32. Chen C, Xu T, Wang G, Li B. Railway turnout system RUL prediction based on feature fusion and genetic programming. *Measurement.* 2020;151:107162.
 33. Mao W, He J, Zuo MJ. Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning. *IEEE Trans. Instrum. Meas.* 2019;69(4):1594-1608.
 34. Zhang D, Qian L, Mao B, Huang C, Huang B, Si Y. A data-driven design for fault detection of wind turbines using random forests and XGboost. *IEEE Access.* 2018;6:21020-21031.
 35. Ecer F, Ardabili S, Band SS, Mosavi A. Training multilayer perceptron with genetic algorithms and particle swarm optimization for modeling stock price index prediction. *Entropy.* 2020; 22(11):1239.
 36. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.2014.
 37. Chin W-S, Zhuang Y, Juan Y-C, Lin C-J. A learning-rate schedule for stochastic gradient methods to matrix factorization. *Pacific-Asia Conference on Knowledge Discovery and Data Mining.* Cham: Springer; 2015:442-455.
 38. Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization. Paper presented at: NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems; December 2014:2177-2185.

39. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding deep learning requires rethinking generalization. *arXiv Preprint arXiv:1611.03530*.2016.
40. Finnoff W, Hergert F, Zimmermann HG. Improving model selection by nonconvergent methods. *Neural Netw.* 1993;6(6):771-783.
41. Fan J, Fan J, Liu F, Qu J, Li R. A novel machine learning method based approach for Li-ion battery prognostic and health management. *IEEE Access.* 2019;7:160043.

How to cite this article: Rouhi Ardehshiri R, Ma C. Multivariate gated recurrent unit for battery remaining useful life prediction: A deep learning approach. *Int J Energy Res.* 2021;1-16. <https://doi.org/10.1002/er.6910>