IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 6, 2019, accepted August 23, 2019, date of publication August 27, 2019, date of current version November 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937798

Deep Learning Prognostics for Lithium-Ion Battery Based on Ensembled Long Short-Term Memory Networks

YUEFENG LIU[1,2], GUANGQUAN ZHAO[1], AND XIYUAN PENG[1]

[1]School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China
[2]School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, China

Corresponding author: Guangquan Zhao (hit53zhao@hit.edu.cn)

This work was supported by the Inner Mongolia Natural Science Foundation under Grant 2018MS06019.

ABSTRACT In recent years, a notable development for predicting the remaining useful life (RUL) of components is prognostics that use data-driven approaches based on deep learning. In particular, long short-term memory networks (LSTMNs) have been successfully applied in RUL prediction. However, to the best of our knowledge, these deep learning-based prognostics do not take into account uncertainty, and their prediction performance needs improvement. Bayesian model averaging (BMA) is a very useful ensemble method because it can quantify uncertainty. In this paper we propose a deep learning ensembled prediction approach based on BMA and LSTMNs. We constructed multiple LSTMN models with different subdatasets derived from the degradation of training data. Then, BMA was used to integrate the LSTMN submodels into one framework for a reliable prognostic. The main advantages of this method are that it 1) provides uncertainty management by postprocess forecast ensembles to create predictive probability density functions (PDFs) and generate probabilistic predictions with uncertainty intervals using BMA and 2) it improves prediction performance by ensemble multiple deep learning submodels (trained with different subdatasets) with corresponding weights calculated by the posterior model probability of the BMA. Finally, we introduced an online iterated training strategy for the BMA algorithm to realize higher prediction performance than that of an offline training strategy. In the experiments, we used lithium-ion battery data sets from the Center for Advanced Life Cycle Engineering at the University of Maryland. The results demonstrate the effectiveness and reliability of our proposed ensemble prognostic approach.

INDEX TERMS Deep learning, LSTMN, BMA, ensemble approach, prognostic.

I. INTRODUCTION

Lithium-ion batteries are widely used in consumer electronics and industrial systems. Remaining useful life (RUL) predictions for lithium-ion batteries allow predictive maintenance to be performed. This is very important for the reliability of these devices, especially for some key electronic devices such as implantable medical devices or satellites and so on.

Various methods for RUL prediction of lithium-ion batteries have appeared in recent years. In general, these approaches can be categorized into 1) model-based approaches, 2) data driven approaches, and 3) hybrid approaches based on 1) and 2). Model-based methods are typically implemented by constructing a physical degradation model that describes the internal electrochemical reactions of a lithium-ion battery. However, for lithium-ion batteries, the physical degradation model is difficult to establish as a dynamic nonlinear system, and the parameters are inconsistent under different working conditions and workloads, which makes parameter identification more difficult. The data driven method estimates RUL based on historical data and monitoring data. Many data-driven methods have been applied to battery RUL estimation. Zhou et al. presented an incremental optimized relevance vector machine (RVM) algorithm to estimate the lithium-ion battery RUL [1]. Zhang et al. proposed an improved unscented particle filter (IUPF) method for lithium-ion battery RUL prediction based on Markov chain Monte Carlo (MCMC) [2]. Song et al. proposed an iterative updated approach with an iterative updated RVM fused with the Kalman filter (KF)

The associate editor coordinating the review of this article and approving it for publication was Chong Leong Gan.

algorithm to improve the long-term prediction performance for a battery's RUL prediction [3]. Hu et al. presented a hybrid method to predict the remaining useful life of lithium-ion batteries in implantable medical devices [4]. Sun et al. developed an integrated health indicator to predict remaining useful life of batteries [5]. Zhang et al. developed a fusion technique consisting of relevance vector machine and particle filter (PF) to construct an aging model of the battery for RUL prediction [6]. Zhang et al. also developed a RUL prediction method based on the Box-Cox transformation (BCT) and Monte Carlo (MC) simulation for battery RUL prediction [7]. Peng et al. proposed a prognostic method fusing the wavelet de-noising (WD) method and the hybrid Gaussian process function regression (HGPFR) model for predicting the RUL of the lithium-ion battery [8]. Zhang et al. proposed a RUL prediction method based on the exponential model and the particle filter for lithium-ion battery [9]. Zhao et al. proposed a prediction method based on the support vector regression for the RUL of lithium-ion batteries [10]. However, these methods mentioned above have shallow architectures, which have insufficient prognostic capability and suffer from curse of dimensionality. In recent years, one notable development in prognostics is the data-driven approach based on deep learning. The advances of deep learning introduce new data-driven approaches to this problem.

Deep learning has been applied successfully in a variety of domains. Its main focus has been in image processing, but recently research has emerged on deep learning for RUL prediction. Ren *et al.* [11] proposed a deep learning-based prediction framework for the RUL of batteries by using a deep autoencoder and deep neural networks. Phattara et al. used a Deep Neural Networks (DNN) approach to predict the RUL of the lithium-ion battery [12], and proved that the deep learning algorithm outperformed other traditional machine learning algorithms. Zhang et al. proposed a LSTM-RNN method for the lithium-ion battery remaining useful life prediction [13]. Liao *et al.* [14] proposed a novel restricted Boltzmann machine for predicting the RUL of systems. Zhang *et al.* [15] proposed an ensemble of deep belief networks (DBNs) for fault diagnosis and in [16] extended this work to estimate the RUL of a mechanical system. Babu *et al.* [17] built a convolutional neural network (CNN) to predict the RUL of a system. Zheng *et al.* [18] studied long short-term memory networks (LSTMNs) for RUL estimation. Li *et al.* [19] proposed a data-driven approach for prognostics using deep CNNs. Zhao *et al.* [20] used an LSTMN to predict RUL associated with tool wear. This work was followed by a proposed design of an integrated architecture using a CNN and an LSTMN [21]. Although various deep learning approaches had been proposed for RUL prediction, the deep learning approach for the RUL prediction of lithium-ion batteries is still limited. To the best of our knowledge these publications on deep learning-based prognostics do not include the study of uncertainty. Uncertainty representation, which is dominated by probability theory, provides a mathematically rigorous approach with statistically sufficient assumptions [22]. Uncertainty management can provide more reliable results with a probability confidence interval for condition-based maintenance and is an important subject in the reliability community. To accommodate uncertainty, in this paper we propose a deep learning-based approach for RUL prediction that includes uncertainty management.

Because different models have their own strengths in capturing different aspects of real-world processes, combining the results from diverse models by weighting procedures can yield a better performance than that of any individual model. Hybrid or fusion models have also been proposed and widely used in lithium-ion battery RUL estimation [23]–[25]. Bayesian model averaging (BMA), a method for averaging results from different competing models, has been used to make ensemble predictions because it can deal with uncertainty. It has been applied in diverse fields such as load demand [26], climate [27], seasonal time series data forecasting [28], energy models [29], prediction of the semiconductor industry cycle [30], wind speed [31], rainfall prediction [32], photovoltaic demand forecasting [33], solar output power forecasting [34], and aerospace [35]. These studies have demonstrated that BMA can obtain much better results than those of any individual model. However, those BMA applications were based on shallow architecture submodels rather than complex deep architecture submodels. Therefore, it is worth investigating whether BMA based on deep learning-based models can produce more accurate results than single deep learning architecture submodels. Furthermore, most of the above BMA algorithms were built with offline models. As a result, the precision of their predictions was lower than that of online algorithms [36]. In our experiments, we used a lithium-ion battery RUL estimation application to demonstrate our proposed approach, but this approach can be extended to other prognostic applications. Therefore, this study had two main goals: First, to include uncertainty management in prognostics by postprocessing forecast ensembles to create predictive probability density functions (PDFs) and generate probabilistic predictions with uncertainty intervals using BMA. The second goal was to improve prediction precision by using ensembles of multiple LSTMN submodels. We calculated the weights of these submodels by modeling the posterior probability of BMA, using offline and online training strategies.

The rest of this paper is organized as follows. First, related work on LSTMNs and BMA are briefly reviewed in Section 2. Section 3 introduces our proposed fusion of BMA with different LSTMN submodels using offline and online training strategies. In Section 4, experiments using lithium-ion battery data sets are described, and the results are given. Conclusions are in Section 5.

II. RELATED WORK

A. LONG SHORT-TERM MEMORY NETWORKS

LSTMNs belong to the family of (deep) RNNs and are capable of addressing the RNN vanishing gradient problem [37].

The key to LSTMNs is the cell state. The cell state is analogous to a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. Information flows very easily along it and remains unchanged. An LSTMN can remove or add information to the cell state using structures called gates. Gates are a way of optionally letting information through. An LSTMN has three of these gates (forget gate, input gate, output gate), to protect and control the cell state. The core idea behind LSTMNs is that these three kinds of gates are used to control the passing of information along sequences that can capture long-range dependencies more accurately. When working over longer periods of time, the gates allow the LSTMN units to read, write, and remove information from memory. This characteristic enables the units to hold only relevant data while "forgetting" irrelevant information. A single localized LSTMN cell in the first layer of a network at time step t is shown in Fig. 1.

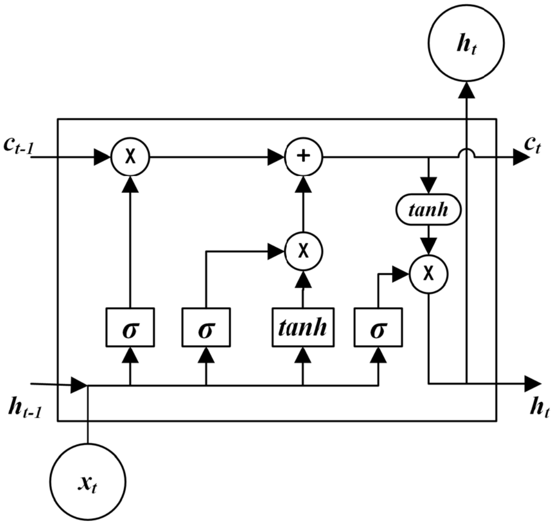


FIGURE 1. Schematic of an LSTMN. c_t is the cell state of time t, c_{t-1} is the cell state of time $t-1$, h_t is the output state (also known as the hidden state) of time t, h_{t-1} is the output state of time $t-1$, and σ is the sigmoid activation function.

In an LSTMN, at each time step t, the hidden state h_t is updated by current data at the same time step x_t, the hidden state at the previous time step h_{t-1}, the input gate i_t, the input node g_t, the forget gate f_t, the output gate o_t, and a memory cell c_t. The updating equations are

$$\begin{aligned} f_t &= \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \\ i_t &= \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + b_i\right) \\ g_t &= \tanh\left(W_{gx}x_x + W_{gh}h_{t-1} + b_g\right) \\ c_t &= g_t * i_t + c_{t-1} * f_t \\ o_t &= \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right) \\ h_t &= \tanh\left(c_t\right) * o_t \end{aligned} \tag{1}$$

where W and b are the layer weights and biases respectively.

Because LSTMNs are able to capture long-range dependencies and nonlinear dynamics in time series data, LSTMNs have been successfully applied in various applications, particularly in speech recognition [38], natural language processing [39], image captioning [40], and RUL prediction [41]. Capacity degradation data, which can cover thousands of charge/discharge cycles and represent the degradation evolution of batteries, can be regarded as long-term time series data. In this study, we used an LSTMN, with its ability to access the previous context of each specific time step, to learn the long-term dependency of the degradation data of capacities. We then used the trained LSTMN model as a potential submodel for BMA to predict the RUL of lithium-ion batteries.

B. BAYESIAN MODEL AVERAGING

BMA is a statistical way of postprocessing forecast ensembles to create predictive PDFs for various applications, as mentioned above. Suppose a linear model structure, the parameter y denotes the dependent variable, X_λ is the explanatory variables, α_λ is a constant, β_λ is the coefficients, ε is a normal IID error term with variance σ^2:

$$y = \alpha_\lambda + X_\lambda\beta_\lambda + \varepsilon \quad \varepsilon \sim N(0, \sigma^2 I) \tag{2}$$

The parameter X denote a matrix that includes many potential explanatory variables, then, a problem arises: Which variables $X_\lambda \in \{X\}$ should be included in the model? And how important are they? The direct approach to do inference on a single linear model that includes all variables is inefficient or even infeasible with a limited number of observations. BMA tackles the problem by estimating models for all possible combinations of {X} and constructing a weighted average over all of them. If X contains K potential variables, this means estimating $2^{\{K\}}$ variable combinations and thus $2^{\{K\}}$ models. The model weights for this averaging stem from the posterior model probabilities that arise from the Bayes' theorem,

$$\begin{aligned} p(M_\lambda|y, X) &= \frac{p(y|M_\lambda, X)p(M_\lambda)}{p(y|X)} \\ &= \frac{p(y|M_\lambda, X)p(M_\lambda)}{\sum_{s=1}^{2^k} p(y|M_s, X)p(M_s)} \end{aligned} \tag{3}$$

where p(y | X) is the integrated likelihood, which is constant over all models and is thus simply a multiplicative term. Therefore, the posterior model probability $p(M_\lambda \mid y, X)$ is proportional to the marginal likelihood of the model $p(y \mid M_\lambda, X)$ (the probability of the data given the model M_λ) times a prior model probability $p(M_\lambda)$; that is, how probable the researcher thinks model M_λ is before looking at the data. Renormalization then leads to the posterior model probability (PMP), and thus the model weighted posterior distribution for any statistic (e.g., the coefficients):

$$p(\theta|y, X) = \sum_{\lambda=1}^{2^k} p(\theta|M_\lambda, y, X)p(M - \lambda|X, y) \tag{4}$$

The model prior $p(M_\lambda)$ must be elicited by the researcher and should reflect prior beliefs. A popular choice is to set a uniform prior probability for each model $\mathrm{p(M_\lambda)} \propto 1$ to represent the lack of prior knowledge.

In our proposed model, the parameter y represents the true RUL value, i.e. the label, and X represents the matrix of RUL prediction results for all LSTMN models.

The LSTMN submodels' weights can be calculated by the posterior model probabilities that arise from the Bayes' theorem in Equation (3). For example, with a small number of variables, it is straightforward to enumerate all potential variable combinations to obtain posterior results. For a larger number of covariates, this becomes more time intensive: enumerating all models for 25 covariates takes about 3 hours on a modern PC, and doing a bit more already becomes infeasible: With 50 covariates for instance, there are more than a quadrillion (10^{15}) potential models to consider. In such a case, Markov Chain Monte Carlo (MCMC) samplers gather results on the most important part of the posterior model distribution and thus approximate it as closely as possible. BMA mostly relies on the Metropolis-Hastings algorithm, which 'walks' through the model space as follows:

At step i, the sampler stands at a certain 'current' model $\mathrm{M_i}$ with PMP $\mathrm{p(M_i|y,X)}$. In step i + 1 a candidate model $\mathrm{M_j}$ is proposed. The sampler switches from the current model to model $\mathrm{M_j}$ with probability $\mathrm{p_{i,j}}$:

$$p_{i,j} = \min(1, p(M_j|y, X)/p(M_i|y, X) \tag{5}$$

In case model $\mathrm{M_j}$ is rejected, the sampler moves to the next step and proposes a new model $\mathrm{M_k}$ against $\mathrm{M_i}$. In case model $\mathrm{M_j}$ is accepted, it becomes the current model and has to survive against further candidate models in the next step. In this manner, the number of times each model is kept will converge to the distribution of posterior model probabilities $\mathrm{p(M_i|y,X)}$.

After getting the submodels' posterior model probabilities, we can get a list keeping all the submodels sorted by their PMP. In our work, X includes 4 potential variables (LSTMN1 to LSTMN4), so there are a total of 2^4 variable combinations (i.e. submodels). The criteria for determining the N of most effective submodels are based on their PMP, some submodels will be removed if their PMP is very small, such as below 0.01, and the rest N submodels will be selected as the best models. Then we chose N of the most effective submodels and constructed the final ensemble models. Suppose each submodel estimation result were f_n (n is in $1 - N$). Then, we used the PMP to divide the sum of those results as the new submodel posterior probability. Next, the new postprobability was named ω_n (n is in $1 - N$): the weight of the submodel. Fig. 2 shows the BMA algorithm constructing LSTMN submodels with different dataset diagrams.

Finally, the ensemble result f_F is given using

$$f_F = \sum_{n=1}^{N} \omega_N f_n \tag{6}$$

More details about the BMA method can be found in [42].

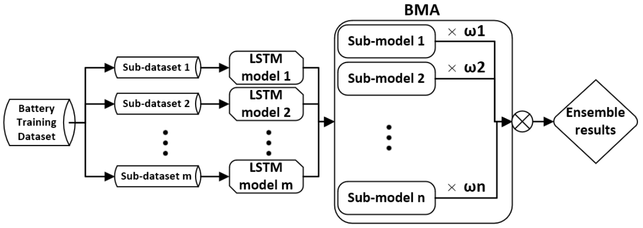


FIGURE 2. BMA algorithm constructing different LSTMN submodels.

III. BMA-LSTMN PREDICTION METHODS FRAMEWORK

This section is organized as follows. First, we describe the overall framework of the proposed approach. Next, we introduce the LSTMN model construction for predicting the RUL of lithium-ion batteries. Then, we present how to ensemble the LSTMN model with offline BMA. Finally, an online BMA method is discussed and compared with the offline algorithm.

A. OVERALL FRAMEWORK

Fig. 2 is an overall framework diagram of our proposed approach. It is known that BMA must select submodels before ensembling the final results. Here, m LSTMN models were chosen to construct the top n best submodels of the BMA. First, m LSTMN models were obtained by m different subdatasets derived from the battery training dataset, and the value of m was a hyperparameter selected based on trial-and-error. Then those m LSTMN models were used as the potential variables of the input vector of the BMA. The BMA was able to estimate the top n best submodels for all possible combinations of the m variables included in the input vector and construct a weighted average of all of them. Details on how to get the top n best submodels and their corresponding weights are in Section II.

The procedure of the proposed prognostics approach is as follows:

Step 1: Obtain m different subdatasets from the lithium-ion battery capacity degradation training dataset, and normalize the subdatasets to [0, 1].

Step 2: Do the training with the m different subdatasets and get m different LSTMN models.

Step 3: Use the m LSTMN models as the m potential variables of the BMA. To obtain the top n best submodels and the corresponding weights ω, use the BMA algorithm with the training dataset for the battery for which the RUL needs to be predicted.

Step 4: Calculate the ensemble RUL results for the lithium-ion battery as in (4).

B. LSTMN MODEL CONSTRUCTION

The core components of an LSTMN model are a sequence input layer and an LSTM layer. A sequence input layer inputs sequence or time series data into the network. An LSTM layer learns long-term dependencies between the time steps of the sequence data. The architecture of an LSTMN model for training is shown in Fig. 3.

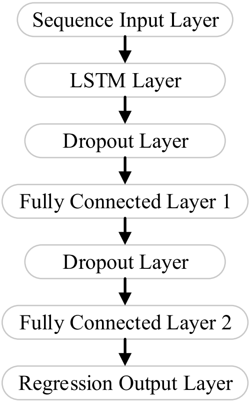


FIGURE 3. Architecture of an LSTMN for training.

The network started with a sequence input layer followed by an LSTM layer. The network ended with a fully connected layer and a regression output layer. Since the deep architecture is more efficient in representation learning, stacking the LSTM layer and the fully connected layer makes sense, but our lithium-ion battery dataset is single-dimensional and the number of samples is limited, and the deep network is prone to overfitting, which will weaken the generalization capabilities of the model. In fact, we tried to build a network with two layers of LSTM in the experiment, which proved that the effect is not as good as that of a network with only one LSTM layer. Therefore, our deep neural network is simple, requiring only one LSTM layer and two fully connected layers, followed by two dropout layers and the final regression output layer. Dropout was introduced during model training [43]. Via the dropout, parts of the hidden outputs were randomly masked so that those neurons would not influence forward propagation during training. During the testing phases, the dropout was turned off, and the outputs of all hidden neurons affected model testing. In our models, the design of dropout layer is similar to that described in [44]. Standard dropout perturbs the recurrent connections, which makes it difficult for the LSTM to learn to store information for long periods of time. By not using dropout on the recurrent connections, the LSTM can benefit from dropout regularization without sacrificing its valuable memorization ability. So, we only dropout on the output of the layer as shown in Fig. 3, we adopted one dropout layer between the LSTMN layer and the first fully connected layer, and another dropout layer between the first fully connected layer and the second fully connected layer. Their masking probabilities were both set to 0.5, as shown in

$$\overline{pred} = relu(dropout((W_{out}h_t + b_{out}), 1)) \tag{7}$$

where W_{out} and b_{out} are the weights and biases of the output. In our tasks, the output value $\overline{pred}$ was the battery capacity. We choose *relu* rather than *sigmoid* as the activation function in accordance with the literature [42]. For model training, we compared the predicted battery capacity value with the true battery capacity value y to obtain the mean squared error as model loss:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{pred_i})^2 \tag{8}$$

where n is the training sample size. We trained the LSTMN with full back propagation through time under the stochastic gradient descent method, as in [45].

C. OFFLINE BMA

In ensemble learning based on BMA, first different datasets are chosen to construct different LSTMN submodels, and then all these LSTMN submodels are combined using BMA. After the LSTMN submodels are constructed, the postprobability of those submodels in the BMA frame should be computed. In addition, to provide more accurate comprehensive model results, the BMA can also provide a qualitative evaluation of the model structure uncertainty. Using the BMA method to combine different LSTMN models can obtain the probability distribution of the prediction results sequence. Among them, the mean value of the distribution can be used as a multiple model integrated result, where the variance and confidence interval of the distribution reflect the uncertainty caused by the LSTMN model structure. The variance of the BMA analog variable posterior probability distribution can be expressed as

$$Var(y|D) = \sum_{k=1}^{k}\omega_k(f_k - \sum_{k=1}^{k}\omega_k f_k)^2 + \sum_{k=1}^{k}\omega_k\sigma_k^2 \tag{9}$$

where σ_2^k is the analog variable variance under the condition of given observation data D and the model f_k. The BMA analog variable variance Var includes the between-model error and the within-model error. In Eq. (7), $\Sigma_{k=1}^{K}\omega_k(f_k - \Sigma_{k=1}^{K}\omega_k f_k)^2$ is the between-model error, and $\Sigma_{k=1}^{K}\omega_k\sigma_k^2$ is the within-model error. Compared with the deterministic multimodel combination method, the uncertainty can be better described by the BMA analog variable variance.

The Monte Carlo method was used to generate BMA probabilistic predictions with uncertainty intervals at every moment. Details can be found in [46]. Each step was sampled 20,000 times to generate the BMA probabilistic ensemble predictions in this study, and then the predictions were sorted in ascending order.

The 90% uncertainty intervals were then derived within the range of 5% and 95% quantiles. We used two indicators (coverage and interval width) to characterize the optimality of the BMA confidence interval. "Coverage" refers to the ratio of the confidence interval coverage to the measured true data; the greater the coverage value, the more accurate is the confidence interval result. Interval width is also one of the commonly used evaluation indicators of confidence interval: for a certain confidence level, the interval width should be as narrow as possible assuming that high coverage is guaranteed.

D. ONLINE ITERATED BMA

For the dynamic variable situation during a load condition, the degradation process of a lithium-ion battery RUL trends toward degradation, but there are obvious local energy regenerations. Currently, most lithium-ion battery RUL prediction algorithms are offline approaches. However, offline modeling uses only the currently known samples in the learning process to estimate the model, and then uses the trained offline model for the consequent new samples to make a multistep prediction; no more modifications are made to the offline model in the process. Thus, directly outputting estimation results with an offline model results in lower precision and does not trace the system variation trend. Therefore, we used some online learning methods to build an effective model.

Basing our work on the offline BMA method, and using the concepts in [36], we used an online iterated BMA approach. When a new measurement sample was obtained, the training set was reconstructed, and m LSTMN models were rebuilt with the new dataset. Then we estimated the lithium-ion battery RUL using the new dynamic BMA algorithm ensembled with the m new LSTMN models. The procedure is described after these definitions:

$DS_1, DS_2, \ldots, DS_\mathrm{m}$ are the original sample sets for the m LSTMN models; NDS is the new increased dataset; $WS_1, WS_2, \ldots, WS_\mathrm{m}$ are the working datasets of the m LSTMN models.

Online BMA algorithm flow:

Step 1: Choose the original training datasets DS_1, $WS_2 = DS_2, \ldots, WS_\mathrm{m} = DS_\mathrm{m}$ and construct m different LSTMN models with the working datasets WS_1, $WS_2, \ldots, WS_\mathrm{m}$ as initial offline BMA model parameters.

Step 2: Use the BMA model ensembled with the m LSTMN submodels to estimate the RUL result.

Step 3: When the new sample NDS is obtained, let $WS_1 = WS_1 + NDS$, $WS_2 = WS_2 + NDS, \ldots, WS_\mathrm{m} = WS_\mathrm{m} + NDS$. Then reconstruct the m new LSTMN models with the updated $WS_1, WS_2, \ldots, WS_\mathrm{m}$.

Step 4: Update the new dynamical BMA model ensembled with the m new LSTMN models.

Step 5: Repeat steps 2 to 4 until the end of prediction.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. EXPERIMENT DATA SETS

The experiment data sets were lithium-ion battery data sets derived from the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland [47]. In this experiment, the batteries were tested to find their capacity degradation. The test was done using the Arbin BT2000 battery testing system (Arbin Instruments, USA) at room temperature. We used experiment datasets with 1.1-Ah rated capacity. Figure 4 shows the capacity degradation curves of No. 35, No. 36, No. 37, and No. 38 lithium-ion batteries. In this experiment, the capacity was selected as the health indicator of the degradation of the lithium-ion battery. When the lithium-ion battery reached its end of life (EOL), the charging capacity reached approximately 70% of the rated capacity. The experiment was then stopped.

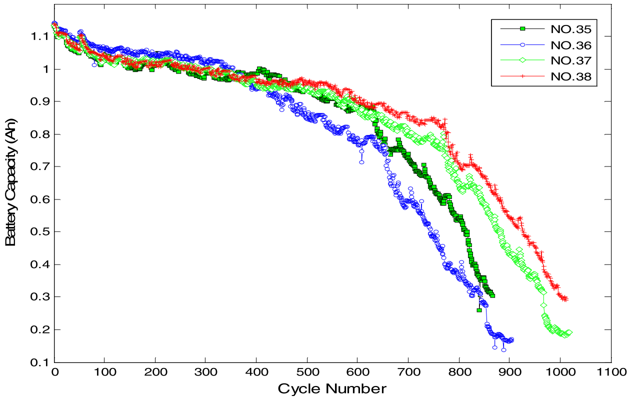


FIGURE 4. Capacity degradation curves of No. 35, No. 36, No. 37 and No. 38 batteries.

B. EXPERIMENT SETTINGS AND EVALUATION CRITERION

For computing hardware, we used two cores of an Intel Xeon CPU at 3.00 GHz and an NVIDIA Quadro M2200 GPU. Because the LSTMN models were built using MATLAB, we did experiments using MATLAB 9.4 release 2018a mixed with R version 3.1.2, but for the BMA model, we used an R package. The data in our experiments is battery capacity, they are time series data with no missing values and outliers. Therefore, the only thing we need to do for data preprocessing is normalization. The data is normalized to [0, 1]. For training an LSTMN, the batch size was set to 50 for the stochastic gradient dissent method using a learning rate of 0.1 and a momentum of 0.9. The stopping criterion was that the mean squared error given by Eq. (6) be below 0.0001, or the number of iterations reached the maximum value. Our LSTMN model has seven layers, the first layer is the sequence input layer, the input size is 1, the second layer is the LSTM layer, there are 39 hidden neurons, the third layer is the dropout layer, and the dropout probability is 0.5, the fourth layer is a fully connected layer with a layer size of 20, the fifth layer is another dropout layer with a dropout probability of 0.5, the sixth layer is also a fully connected layer with an output size of 1, and the last layer is a regression layer with a layer size of 1 to calculate the MSE loss. The common parameters of each LSTMN model used are shown in Table 1. Note that these parameters were selected based on trial-and-error.

The battery capacity degradation data can be seen as a time series with one feature (input size = 1). A flow of length 39 (time step = 39) time series was passed once through the LSTM layer. This means that a time series sample consisting of $\{x_i, x_{i+1}, \cdots, x_{i+39}\}$ was used as the sequence input layer of the LSTMN, and the output $\bar{x}_{i+40}$ of the LSTMN was the prediction value of the capacity in the next moment of the time series, where x_i is the actual battery capacity value of the ith cycle.

We did a series of experiments with several battery data sets. Here, only the RUL estimation results of the

TABLE 1. Parameters used in each LSTMN model.

Parameters	Settings
Input size	1
Time steps	39
Batch size	50
Dropout	0.5
The first fully connected layer size	20
The second fully connected layer size	1
Regression layer size	1
Learning rate	0.1
Momentum	0.9

TABLE 2. Teaining data sets in each LSTMN model and BMA-LSTMN.

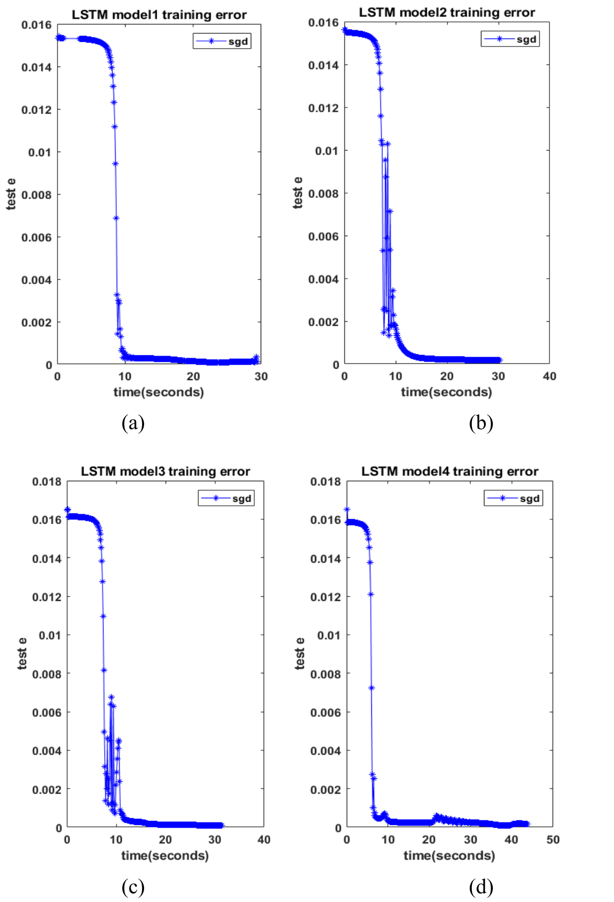
model	training data set
LSTMN model 1	battery No. 35 and No. 36
LSTMN model 2	battery No. 35 and No. 38
LSTMN model 3	battery No. 36 and No. 38
LSTMN model 4	battery No. 35, No. 36 and No. 38
BMA-LSTMN	the prediction results by LSTMN model 1 to LSTMN model 4 on battery No. 37

No. 37 battery is provided to briefly explain the experimental process. The No. 35, No. 36, and No. 38 battery datasets were used as training datasets. We chose four different subdatasets as the training samples for four LSTMN models by reconstructing the No. 35, No. 36 and No. 38 battery datasets. Taking the LSTMN model 1 as an example, the training sample was a combination of the No. 35 battery and No. 36 battery datasets. The combination method was simply to link those two datasets head-to-tail. The training sample of LSTMN model 2 was the combination of the No. 35 battery and No. 38 battery datasets. The training sample of LSTMN model 3 was a combination of the No. 36 battery and No. 38 battery datasets. The training sample of LSTMN model 4 was the combination of the No. 35, No. 36, and No. 38 battery datasets. The training data set of BMA-LSTMN is the prediction results from LSTMN model 1 to LSTMN model 4 on battery No. 37. The differences of the training data sets between the LSTMN models and the BMA-LSTMN are shown in Table 2.

The training errors of the four LSTMN models are shown in Fig. 5.

After the LSTMN models were trained, we could predict the RUL of the No. 37 battery using each LSTMN model individually. There were a total of 800 charge/discharge cycles in the No. 37 battery datasets. If the forecast starting point (SP) were set to 700, the prediction process would be

$$\begin{cases} \{x_{662}, x_{663}, \cdots, x_{699}, x_{700}\} \rightarrow \bar{x}_{701} \\ \{x_{663}, x_{664}, \cdots, x_{700}, \bar{x}_{701}\} \rightarrow \bar{x}_{702} \\ \{x_{664}, x_{665}, \cdots, \bar{x}_{701}, \bar{x}_{702}\} \rightarrow \bar{x}_{703} \\ \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ \{\bar{x}_{761}, \bar{x}_{762}, \cdots, \bar{x}_{798}, \bar{x}_{799}\} \rightarrow \bar{x}_{800} \end{cases} \tag{10}$$



(a) (b)

(c) (d)

FIGURE 5. Four LSTMN model training processes of No. 35, No. 36, No. 37 and No. 38 batteries. (a) Model1. (b) Model2. (c) Model3. (d) Model4.

while the evaluation criterion of RUL prediction is the RUL predicted error, which is used to measure the performance of the proposed method as follows:

RUL_{error}: RUL predicted errors:

$$RUL_{\text{error}} = \left|RUL_{pred} - RUL_{true}\right| \tag{11}$$

where RUL_{pred} is the predicted RUL value using the prediction method. To get RUL_{pred}, we need to get the ending point of prediction (EOP) and the SP of prediction. The EOP is the intersection point of two lines (the failure threshold line and the predicted curve); then, $RUL_{pred} = \text{EOP} - \text{SP}$. The actual RUL value is RUL_{true}. Similar to getting the RUL_{pred}, to get the RUL_{true}, we need to get the EOL and the SP. The EOL is at the intersection of two lines (the failure threshold line and the true battery capacity degradation curve), then RUL_{true} = EOL – SP. The RUL computing process is shown in Fig. 6.

C. RUL ESTIMATION BY COMPARING BMA–LSTMN WITH THE SINGLE LSTMN MODELS

1) EXPERIMENT METHOD

Here, the RUL estimation results of the No. 37 battery is provided to briefly explain the results of our experiments

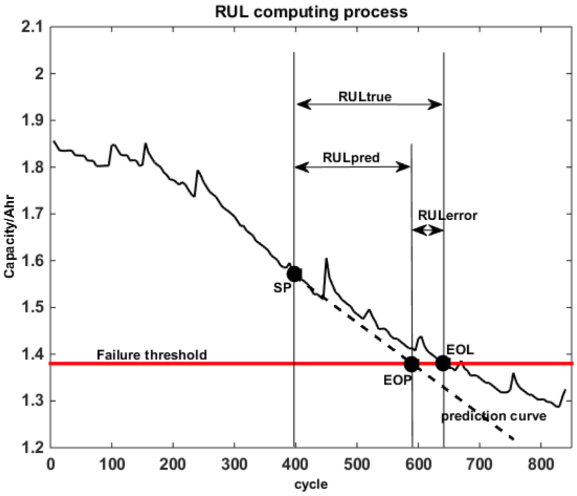


FIGURE 6. RUL computing process.

based on our proposed BMA–LSTMN approach and the single LSTMN models with different datasets. Eight hundred data samples were divided into two data sets: training data samples and testing data samples. The length of the two data sets varied according to the different forecasting SPs. For example, a forecasting SP of 200 denotes that the first 1 to 200 data samples were training data samples, and the remaining 600 data samples were used as testing data samples. In this section, we use an SP of 290 as the first forecasting starting point of the BMA–LSTMN.

As mentioned above, four LSTMN models were trained by four different datasets, and are used here as the input variables for the BMA model. After the BMA model was trained by the No. 37 battery training data sample, a few of the "best" submodels could be gotten. Then the sum of the PMP of these submodels was calculated, and each submodel PMP was divided by the sum to get the weight of the submodel. Finally, the ensemble result was obtained by (4).

2) PREDICTION VALUES OF BMA-LSTMN COMPARED WITH THOSE OF THE SINGLE LSTMN MODELS

The BMA–LSTMN RUL prediction results for the lithium-ion No. 37 battery compared with those of the single LSTMN models at SP = 300, SP = 400, and SP = 500 are shown in Figs. 7, 8, and 9 and Table 3. In Fig. 7, for example, there are five battery capacity degradation curves as compared with the ground truth predicted by LSTMN1, LSTMN2, LSTMN3, LSTMN4, and our proposed BMA–LSTMN model at SP = 300. Clearly, the curve predicted by BMA–LSTMN fits better with the real curve, and the real curve falls in the confidence interval between the upper and lower limit curves, as shown in Figs. 8 and 9. In Table 3, *SP* is the starting point, RUL_{pred} is the RUL prediction value, *CI* is the confidence interval, and RUL_{error} is the evaluation criterion mentioned previously. Comparing the BMA–LSTMN estimation results with those of the single LSTMN models shows that the fusion estimation values by BMA–LSTMN are much more accurate than those of any of the single LSTMN models.

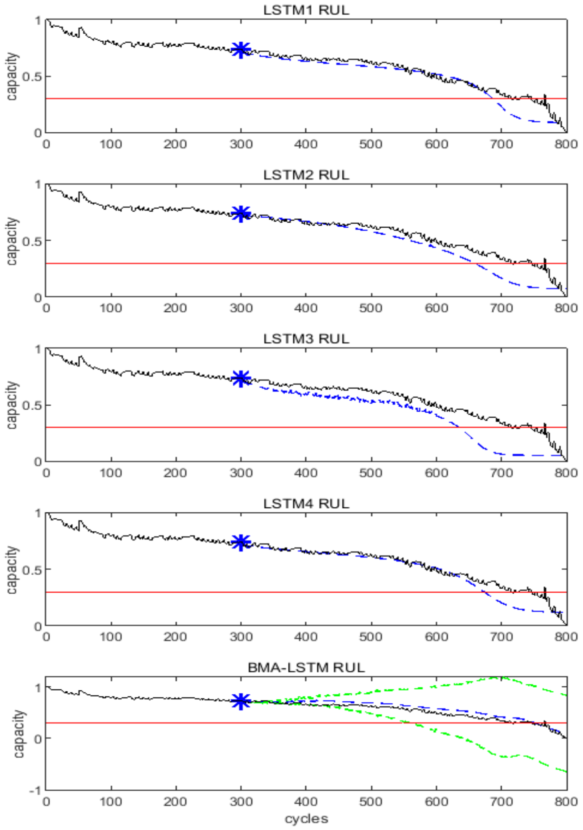


FIGURE 7. BMA–LSTMN RUL prediction results of lithium-ion No. 37 battery compared with those of the single LSTMN models at SP = 300.

The above comparison of prediction accuracy was made only at the 300th, 400th, and 500th cycles. To illustrate the improvement of prediction accuracy over the entire life of the battery, we did a comparison of prediction accuracy in terms of $\alpha - \lambda$ metrics [22], which are defined as

$$[1-\alpha]\cdot\gamma_t(t_k) \le \gamma^l(t_k) \le [1+\alpha]\cdot\gamma_t|(t_k) \qquad (12)$$

where γ^l is the predicted RUL at the lth time instant, γ_t is the ground truth RUL, and α is the accuracy modifier [22].

Fig. 10 shows the $\alpha - \lambda$ metrics with $\alpha = 0.3$ for the prognostic results, and that the predicted RULs are inside the accuracy zone. Compared with the single LSTMN models, the proposed BMA–LSTMN method can obtain much more accurate RUL results. Note that the first SPs of the four LSTMN models were at the 90th cycle, whereas the first SP of the BMA–LSTMN was at the 290th cycle. This is because the BMA model needed to be trained by the four LSTMN submodels' prediction results before being used for final prediction. Here, we set the first training length of the BMA at 200, so the first SP of the BMA–LSTMN can be calculated as 90 + 200 = 290.

D. RUL ESTIMATION BASED ON ONLINE BMA COMPARED WITH OFFLINE BMA

Offline BMA methods had prediction performance better than other methods mentioned above. However, Table 3 shows that the prediction precision of offline BMA was not

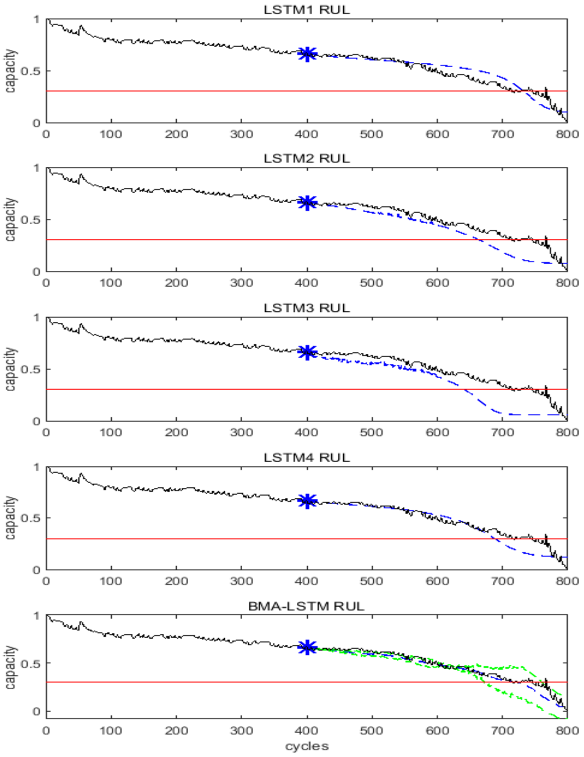


FIGURE 8. RUL computing process BMA–LSTMN RUL prediction results of lithium-ion No. 37 battery compared with those of the single LSTMN models at SP = 400.

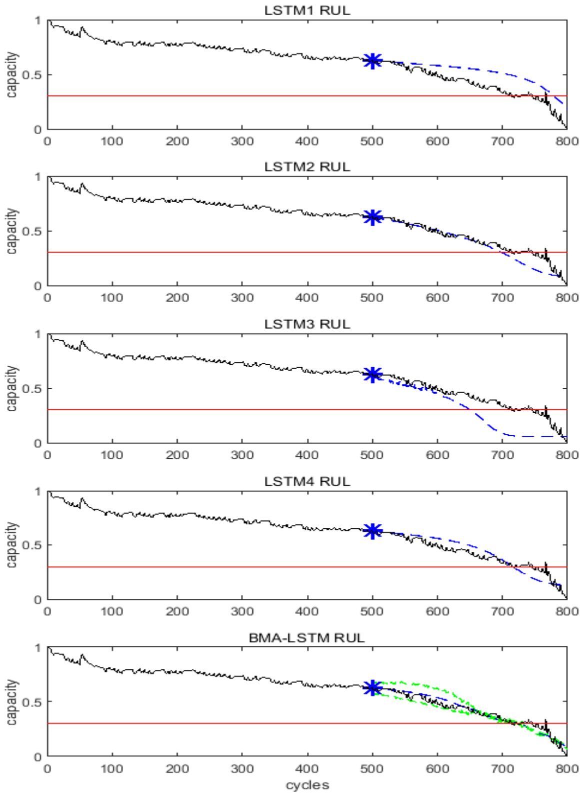


FIGURE 9. BMA–LSTMN RUL prediction results of lithium-ion No. 37 battery compared with those of the single LSTMN models at SP = 500.

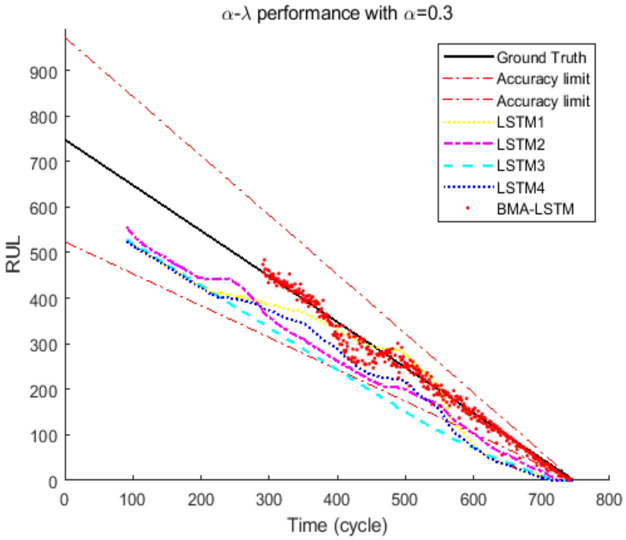


FIGURE 10. BMA–LSTMN RUL prediction accuracy compared with that of the single LSTMN RULs in terms of $\alpha - \lambda$ metrics.

so high. This was due to the offline modeling, which used the currently available sample to train the model only once, then used the offline model to predict the multistep for the consequent samples without modifications. Therefore, this section compares the results of the online learning approach with those of the offline model.

The final ensemble online BMA RUL prediction results compared with the offline BMA for the lithium-ion No. 37 battery at SP = 300, SP = 400, and SP = 500 are shown in Figs. 11–13 and Table 4. To illustrate the improvement of prediction accuracy over the entire life of the battery, a comparison of prediction accuracy done in terms of $\alpha - \lambda$ metrics is shown in Fig. 14. The experiment results show that the online BMA approach can achieve higher precision. The interval widths of 90% CI at three SPs are all narrower than those achieved by offline BMA. This is due to the new samples and the iterated online model training with these new samples. However, it cannot be ignored that the online algorithm is more time consuming because of the iterated online model training in each loop, especially when the model is as complex as our proposed BMA–LSTMN.

E. UNCERTAINTY ANALYSIS OF BMA–LSTMN

Tables 3 and 4 show the 90% CIs of the offline and online BMA–LSTMNs respectively. We can calculate that all the true values at three different SPs are included in their corresponding 90% CI. For example, in Table 3, the true RUL value forecast for an SP of 300 was 447 cycles. The EOL of the No. 37 battery was the 747th cycle, so the true RUL value at an SP of 300 could be calculated as 747 − 300 = 447. This fell within its prediction confidence interval, which was 401,523, and the RUL true value of 347 cycles at an SP of 400 fell within its prediction confidence interval of 302,366. The RUL true value of 247 cycles at an SP of 500 also fell within its prediction confidence interval of 238,280, the same as the

TABLE 3. Quantitative results comparison.

Algorithm	Quantitative results[a]			
	SP	RUL_{pred}	90% CI	RUL_{error}
LSTMN1	300	387	--	60
LSTMN2		358	--	89
LSTMN3		335	--	112
LSTMN4		373	--	74
BMA–LSTMN		**462**	**(401,523)**	**15**
LSTMN1	400	333	--	14
LSTMN2		264	--	83
LSTMN3		243	--	104
LSTMN4		288	--	59
BMA–LSTMN		**334**	**(302,366)**	**13**
LSTMN1	500	280	--	33
LSTMN2		200	--	47
LSTMN3		151	--	96
LSTMN4		216	--	31
BMA–LSTMN		**259**	**(238,280)**	**12**

[a] Quantitative results for lithium-ion No. 37 battery (at different forecasting starting points) based on BMA–LSTMN and single LSTMN algorithms

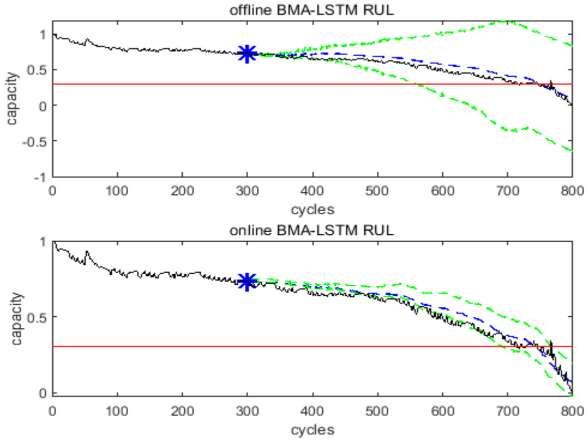


FIGURE 11. Offline BMA–LSTMN RUL prediction result for a lithium-ion No. 37 battery compared with online BMA–LSTMN models at SP=300.

result in Table 4. From the 90% CI of battery capacity prediction values in Figs. 11 to 13 at three different SPs, we can also see that the coverage of the 90% confidence interval is very high—almost 100% coverage—and the interval width was reduced when the SP became larger in both offline and online BMA–LSTMN methods. However, compared with the offline BMA–LSTMN approach, the interval width of the online BMA–LSTMN method was obviously narrower under the condition of a certain confidence level at the same SP. This was due to the new samples and the iterated online model training with those new samples.

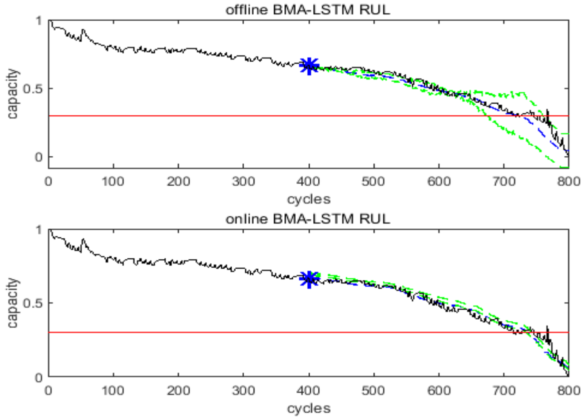


FIGURE 12. Offline BMA–LSTMN RUL prediction result for a lithium-ion No. 37 battery compared with online BMA–LSTMN models at SP=400.

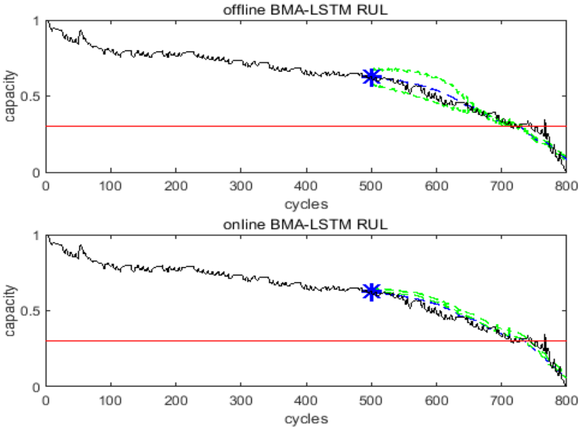


FIGURE 13. Offline BMA–LSTMN RUL prediction result for a lithium-ion No. 37 battery compared with online BMA–LSTMN models at SP=500.

TABLE 4. Quantitative results comparison.

Algorithm	Quantitative results			
	SP	RUL_{pred}	90% CI	RUL_{error}
Offline BMA–LSTMN	300	462	(401,523)	15
Online BMA–LSTMN		**433**	**(411,455)**	**14**
Offline BMA–LSTMN	400	334	(302,366)	13
Online BMA–LSTMN		**337**	**(322,352)**	**10**
Offline BMA–LSTMN	500	259	(238,280)	12
Online BMA–LSTMN		**238**	**(223,253)**	**9**

Figs. 15 to 17 show the probability density function distribution of the estimated battery capacity when reaching the failure threshold (0.3) at SP = 300, SP = 400, and SP = 500. The estimated values clearly are concentrated around the failure threshold.

The best n submodels included in BMA at the above three SPs are also charted in Figs. 18 to 20. In those figures, the blue areas correspond to a positive coefficient, red to a negative coefficient, and white to noninclusion (a zero coefficient). The x-axis shows the best models, scaled by their PMPs. In Fig. 18 for example, two best submodels are included in the

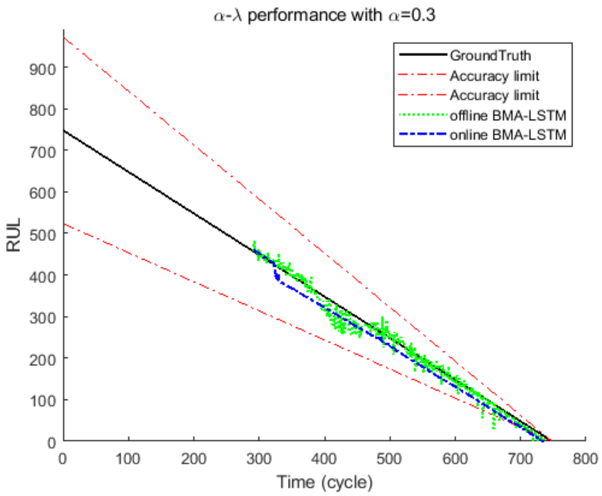


FIGURE 14. Offline BMA–LSTMN RUL prediction result of a lithium-ion No. 37 battery compared with online BMA–LSTMN models in terms of $\alpha - \lambda$ metrics.

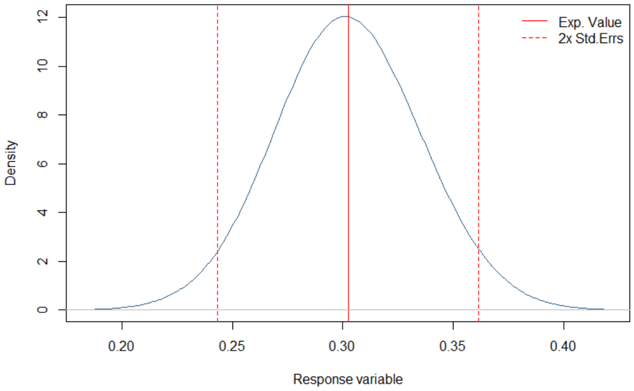


FIGURE 15. Probability density function distribution of the estimated battery capacity when reaching the failure threshold (0.3) at SP = 300.

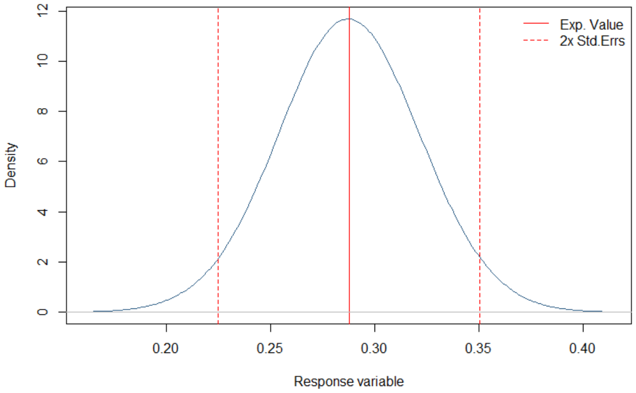


FIGURE 16. Probability density function distribution of the estimated battery capacity when reaching the failure threshold (0.3) at SP = 400.

BMA. The first best submodel includes all four LSTMN models with three positive coefficients for LSTMN1, LSTMN3, LSTMN4, and a negative coefficient for LSTMN2. In the second-best submodel, LSTMN3 is not included. Clearly, the weights of these two best submodels are very different; one is 0.94, and another is $1 - 0.94 = 0.06$.

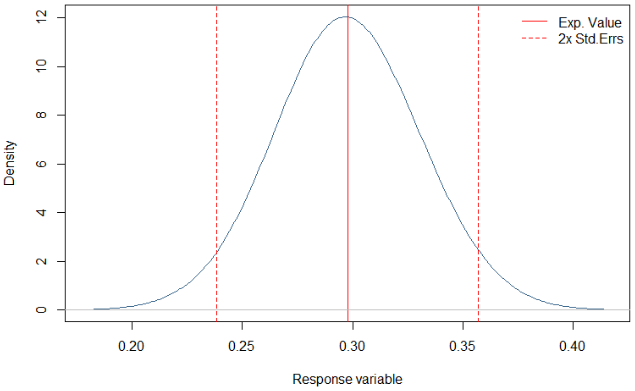


FIGURE 17. Probability density function distribution of the estimated battery capacity when reaching the failure threshold (0.3) at SP = 500.

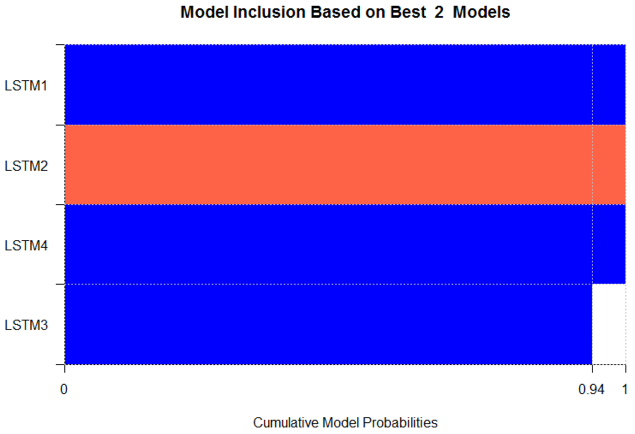


FIGURE 18. Best n models included in BMA at SP = 300.

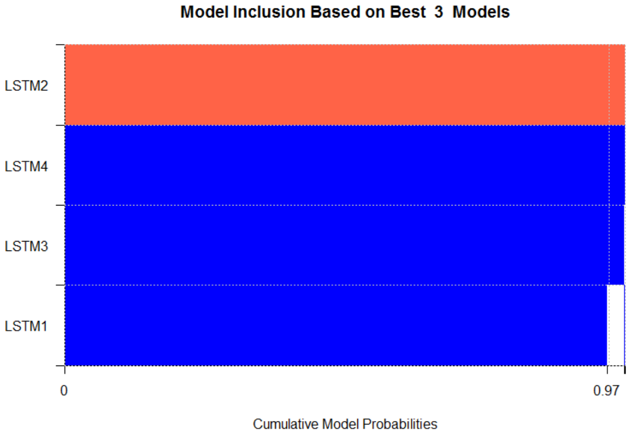


FIGURE 19. Best n models included in BMA at SP = 400.

F. RESULTS COMPARED WITH EXISTING WORKS UNDER THE SAME CONDITIONS

For a valid comparison, we compared the method proposed in [48], [49] using the same experimental data set (No. 37 battery datasets in CALCE) with the BMA–LSTMN method in our study. To simplify the representation, the methods in [48] and [49] were defined as M2 and M3 respectively. The method we named BMA–LSTMN used for comparison

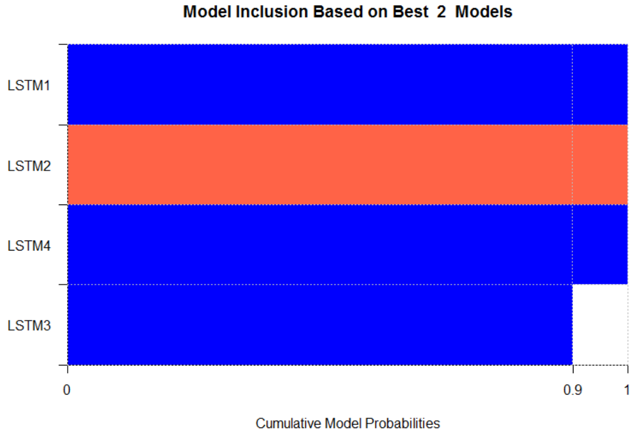


FIGURE 20. Best n models included in BMA at SP = 500.

here was the offline BMA–LSTMN. The evaluation criteria adopted by different documents are different; however, for convenience we used the RUL_{error} defined in Eq. (9) as the evaluation metric for comparison between M2 and our method, and used relative accuracy (RA) defined in Eq. (11) according to [49] as the evaluation metric for comparison between M3 and our method.

$$RA = 1 - \frac{|RUL_{true} - RUL_{pred}|}{RUL_{true}} \tag{13}$$

RA is the measure of the normalized error values between predicted and true RUL. The experimental comparison results of M2 and BMA–LSTMN are shown in Table 5, and results of M3 and BMA-LSTMN are shown in Table 6.

TABLE 5. Quantitative results comparison.

	Quantitative results			
Algorithm	SP	RUL_{pred}	90% CI	RUL_{error}
M2	60 × 5	27× 5	--	55 × 5
BMA–LSTMN	300	**462**	**(401,523)**	**15**
M2	80 × 5	37 × 5	--	25 × 5
BMA–LSTMN	400	**334**	**(302,366)**	**13**
M2	100×5	45×5	--	3×5
BMA–LSTMN	500	**259**	**(238,280)**	**12**

TABLE 6. Quantitative results comparison.

	Quantitative results	
Algorithm	SP	*RA*
M3	120	0.929
BMA–LSTMN	120	**0.966**
M3	240	0.951
BMA–LSTMN	240	**0.962**
M3	360	0.820
BMA–LSTMN	360	**0.923**

Note that a subtraction preprocessing was done on the data in the algorithm of M2, and one new sample was sampled at intervals of 5 points in the original data sample, so the preprocessed data had to be multiplied by 5 to express the real data in Table 5. One can see that the proposed BMA–LSTMN algorithm outperformed the algorithm of M2 and M3.

V. CONCLUSION

In this paper, we propose a deep learning approach with uncertainty management for RUL estimation of lithium-ion batteries. The proposed BMA–LSTMN method ensembled several LSTMN submodels trained by different subdatasets to increase estimation accuracy and provide uncertainty representation results by BMA. The experiments were done on CALCE datasets from the University of Maryland. The results show that the proposed BMA–LSTMN method can provide better RUL estimates than those of the separate individual LSTMN model. It also achieved better results compared with existing methods under the same experiment conditions using the same datasets. Furthermore, the proposed BMA–LSTMN method is capable of uncertainty management. Compared with offline BMA, online BMA had greater RUL prediction precision and less uncertainty for the predicted RUL. In future work, we will ensemble more deep learning approaches to improve prediction performance.

REFERENCES

[1] D. Liu, J. Zhou, D. Pan, Y. Peng, and X. Peng, "Lithium-ion battery remaining useful life estimation with an optimized relevance vector machine algorithm with incremental learning," *Measurement*, vol. 63, pp. 143–151, Mar. 2015.

[2] X. Zhang, Q. Miao, and Z. Liu, "Remaining useful life prediction of lithium-ion battery using an improved UPF method based on MCMC," *Microelectron. Rel.*, vol. 75, pp. 288–295, Aug. 2017.

[3] Y. Song, D. Liu, Y. Hou, J. Yu, and Y. Peng, "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, Jan. 2018.

[4] C. Hu, H. Ye, G. Jain, and C. Schmidt, "Remaining useful life assessment of lithium-ion batteries in implantable medical devices," *J. Power Sources*, vol. 375, pp. 118–130, Jan. 2018.

[5] Y. Sun, X. Hao, M. Pecht, and Y. Zhou, "Remaining useful life prediction for lithium-ion batteries based on an integrated health indicator," *Microelectron. Rel.*, vols. 88–90, pp. 1189–1194, Sep. 2018.

[6] Y. Zhang, R. Xiong, H. He, and M. Pecht, "Validation and verification of a hybrid method for remaining useful life prediction of lithium-ion batteries," *J. Cleaner Prod.*, vol. 212, pp. 240–249, Mar. 2019.

[7] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Lithium-ion battery remaining useful life prediction with Box–Cox transformation and Monte Carlo simulation," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1585–1597, Feb. 2019.

[8] P. Yu, Y. Hou, Y. Song, J. Pang, and D. Liu, "Lithium-ion battery prognostics with hybrid Gaussian process function regression," *Energies*, vol. 11, no. 6, p. 1420, 2018.

[9] L. Zhang, Z. Mu, and C. Sun, "Remaining useful life prediction for lithium-ion batteries based on exponential model and particle filter," *IEEE Access*, vol. 6, pp. 17729–17740, 2018.

[10] Q. Zhao, X. Qin, H. Zhao, and W. Feng, "A novel prediction method based on the support vector regression for the remaining useful life of lithium-ion batteries," *Microelectron. Reliab.*, vol. 85, pp. 99–108, Jun. 2018.

[11] 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. 50587–50598, 2018.

[12] P. Khumprom and N. Yodo, "A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm," *Energies*, vol. 12, no. 4, p. 660, Jan. 2019.

[13] Y. Zhang, R. Xiong, H. He, and Z. Liu, "A LSTM-RNN method for the lithuim-ion battery remaining useful life prediction," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM-Harbin)*, Jul. 2017, pp. 1059–1062.

[14] L. Liao, W. Jin, and R. Pavel, "Enhanced restricted boltzmann machine with prognosability regularization for prognostics and health assessment," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7076–7083, Nov. 2016.

[15] C. Zhang, J. H. Sun, and K. C. Tan, "Deep belief networks ensemble with multi-objective optimization for failure diagnosis," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2016, pp. 32–37.

[16] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, Oct. 2017.

[17] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, Dallas, TX, USA, vol. 9642, Apr. 2016, pp. 214–228.

[18] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2017, pp. 88–95.

[19] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018.

[20] R. Zhao, J. Wang, R. Yan, and K. Mao, "Machine health monitoring with LSTM networks," in *Proc. 10th Int. Conf. Sens. Technol. (ICST)*, Nov. 2016, pp. 1–6.

[21] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional Bi-directional LSTM networks," *Sensors*, vol. 17, no. 2, p. E273, Jan. 2017.

[22] W. Yan, B. Zhang, G. Zhao, J. Weddington, and G. Niu, "Uncertainty management in Lebesgue-sampling-based diagnosis and prognosis for lithium-ion battery," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8158–8166, Oct. 2017.

[23] D. Liu, Y. Luo, J. Liu, Y. Peng, L. Guo, and M. Pecht, "Lithium-ion battery remaining useful life estimation based on fusion nonlinear degradation AR model and RPF algorithm," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 557–572, Sep. 2014.

[24] Y. Zhou and M. Huang, "Lithium-ion batteries remaining useful life prediction based on a mixture of empirical mode decomposition and ARIMA model," *Microelectron. Rel.*, vol. 65, pp. 265–273, Oct. 2016.

[25] Y. Chang, H. Fang, and Y. Zhang, "A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery," *Appl. Energy*, vol. 206, pp. 1564–1578, Nov. 2017.

[26] S. Hassan, A. Khosravi, and J. Jaafar, "Bayesian model averaging of load demand forecasts from neural network models," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 3192–3197.

[27] J. Yang, G. Fang, Y. Chen, and P. De-Maeyer, "Climate change in the Tianshan and northern Kunlun Mountains based on GCM simulation ensemble with Bayesian model averaging," *J. Arid Land*, vol. 9, no. 4, pp. 622–634, Aug. 2017.

[28] A. Vosseler and E. Weber, "Forecasting seasonal time series data: A Bayesian model averaging approach," *Comput. Statist.*, vol. 33, no. 4, pp. 1733–1765, Dec. 2018.

[29] M. Culka, "Uncertainty analysis using Bayesian model averaging: A case study of input variables to energy models and inference to associated uncertainties of energy scenarios," *Energy, Sustainability Soc.*, vol. 6, no. 1, p. 7, Dec. 2016.

[30] W.-H. Liu and S.-S. Weng, "On predicting the semiconductor industry cycle: A Bayesian model averaging approach," *Empirical Econ.*, vol. 54, no. 2, pp. 673–703, Mar. 2018.

[31] S. S. Eide, J. B. Bremnes, and I. Steinsland, "Bayesian model averaging for wind speed ensemble forecasts using wind speed and direction," *Weather Forecast.*, vol. 32, no. 6, pp. 2217–2227, 2017.

[32] P. Vogel, P. Knippertz, A. H. Fink, A. Schlueter, and T. Gneiting, "Skill of global raw and postprocessed ensemble predictions of rainfall over northern tropical Africa," *Weather Forecast.*, vol. 33, no. 2, pp. 369–388, 2017.

[33] M. Q. Raza, M. Nadarajah, and C. Ekanayake, "Demand forecast of PV integrated bioclimatic buildings using ensemble framework," *Appl. Energy*, vol. 208, pp. 1626–1638, Dec. 2017.

[34] M. Q. Raza, N. Mithulananthan, and A. Summerfield, "Solar output power forecast using an ensemble framework with neural predictors and Bayesian adaptive combination," *Sol. Energy*, vol. 166, pp. 226–241, May 2018.

[35] H.-W. Wang, J. Gao, and H.-Q. Wu, "Reliability analysis on aero-engine using Bayesian model averaging," *J. Aerosp. Power*, vol. 29, no. 2, pp. 305–313, Feb. 2014.

[36] J. Zhou, D. Liu, Y. Peng, and X. Peng, "An optimized relevance vector machine with incremental learning strategy for lithium-ion battery remaining useful life estimation," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, vol. 63, May 2013, pp. 561–565.

[37] J. F. Kolen and S. C. Kremer, "Gradient flow in recurrent nets: The difficulty of learning longterm dependencies," in *A Field Guide to Dynamical Recurrent Networks*. 2009.

[38] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[39] S. Wang and J. Jiang, "Learning natural language inference with LSTM," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jan. 2015, pp. 1442–1451.

[40] B. Rister and D. Lawson, "Image captioning with attention," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jan. 2016, pp. 1–5.

[41] M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network," in *Proc. IEEE Int. Conf. Aircr. Utility Syst. (AUS)*, Oct. 2016, pp. 135–140.

[42] C. T. Volinsky, A. E. Raftery, D. Madigan, and J. A. Hoeting, "Bayesian model averaging: A tutorial," *Stat. Sci.*, vol. 14, no. 4, pp. 382–417, 2002.

[43] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*. [Online]. Available: https://arxiv.org/abs/1207.0580

[44] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*. [Online]. Available: https://arxiv.org/abs/1409.2329

[45] Q. Lyu and J. Zhu, "Revisit long short-term memory: An optimization perspective," in *Proc. NIPS*, 2014, pp. 1–9.

[46] L. Dong, L. Xiong, and K. Yu, "Uncertainty analysis of multiple hydrologic models using the Bayesian model averaging method," *J. Appl. Math.*, vol. 2013, Nov. 2013, Art. no. 346045.

[47] 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, Dec. 2011.

[48] Y. Liu, G. Zhao, and X. Peng, "A fusion prediction method of lithium-ion battery cycle-life," *Chin. J. Sci. Instrum.*, vol. 36, no. 7, pp. 1462–1469, Jul. 2015.

[49] K. Pugalenthi and N. Raghavan, "A holistic comparison of the different resampling algorithms for particle filter based prognosis using lithium ion batteries as a case study," *Microelectron. Rel.*, vol. 91, pp. 160–169, Dec. 2018.

YUEFENG LIU received the B.S. degree from the Inner Mongolia University of Technology, Huhhot, China, in 2001, and the M.S. degrees from the Inner Mongolia University of Science & Technology, Baotou, China, in 2008, where he has been an Associate Professor, since 2011. He is currently pursuing the Ph.D. degree from the Harbin Institute of Technology, Harbin, China.

His research interests include prognostics and health management, and deep leaning.

GUANGQUAN ZHAO received the B.E., M.E., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 2000, 2002, and 2007, respectively, where he is currently an Associate Professor and a Master Student Supervisor.

His main research interest includes deep learning and its application on fault diagnosis and prognosis.

XIYUAN PENG received the B.E., M.E., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1984, 1987, and 1992, respectively, where he is currently a Professor and a Ph.D. Candidate Supervisor.

His main research interests include automatic test technology and intelligent fault diagnosis.