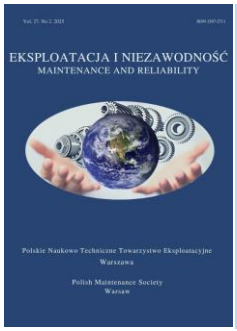


Eksploatacja i Niezawodnosc – Maintenance and Reliability

Volume 27 (2025), Issue 2

journal homepage: http://www.ein.org.pl

Article citation info:
Li J, Ding P, Application of DBN-based KRLS method for RUL prediction of lithium-ion batteries, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2025: 27(2) http://doi.org/10.17531/ein/194174

Application of DBN-based KRLS method for RUL prediction of lithium-ion batteries

Indexed by: Web of Science Group

Jun Li[a,*], Pengfei Ding[a]

[a] School of Automation and Electrical Engineering, Lanzhou Jiao tong University, Lanzhou, Gansu 730070, P.R. China

Highlights

- Using DBN for lithium battery feature extraction solves the problem of difficulty in ex-tracting features from lithium batteries.
- By adopting the concept of dual alternate learning, an SCKF-FB-KRLS fusion model is proposed for RUL prediction of the battery.
- By combining the historical capacity data of lithium batteries, results can be obtained quick-ly and accurately.

Abstract

The Remaining Useful Life (RUL) of lithium batteries is vital for maintaining and safely operating the batteries, making precise RUL predictions highly significant. This paper introduces a method for predicting the RUL of lithium-ion batteries, utilizing a kernel adaptive filtering algorithm integrated with Deep Belief Networks (DBN). The method constructs a novel prediction model based on the Fixed-Budget Kernel Recursive Least Squares (FB-KRLS) algorithm. In this approach, the DBN extracts features from the original lithium battery data to reduce data complexity. The Square-root Cubature Kalman Filter (SCKF) is integrated with the FB-KRLS algorithm, employing a dual alternating learning strategy to improve the model's nonlinear fitting performance. The model was validated using NASA's lithium battery data, showing that the minimum values for the MAPE, RMSE and MAE were 0.102%, 0.0016 and 0.0014, respectively. Therefore, the proposed method demonstrates potential for application in predicting the RUL of lithium-ion batteries.

Keywords
lithium batteries, RUL prediction, FB-KRLS algorithm, SCKF algorithm

1. Introduction

In recent years, libs are extensively employed in consumer electronics, electric vehicles, power storage systems, aerospace, and various other fields owing to their advantages, including low production cost, high energy density, long service life, compact size, and lightweight[1-3]. Battery capacity is an important metric for assessing battery performance. The life of a battery is considered to end when its State of Health (SOH) parameter falls to 70%. Therefore, SOH, which is related to battery capacity, has become the most commonly used indicator for evaluating battery performance[4-5].

Methods for forecasting the Remaining Useful Life of libs can generally be categorized into three types: model-based methods, data-driven methods, and hybrid prediction methods. The effectiveness of model-based approaches is significantly affected by the accuracy of the models constructed. In contrast to model-based methods, data-driven approaches provide more flexibility and practicality, and do not depend on the precision of physical battery models. Instead, they utilize actual battery measurement data to establish models for forecasting the RUL[6-8]. Artificial intelligence methods are commonly used to establish prediction models, including Long Short-Term Memory (LSTM) neural networks[9-10], Support Vector

(*) Corresponding author.
E-mail addresses: J. Li (ORCID: 0000-0002-1406-2734) lijun691201@mail.lzjtu.cn, P. Ding (ORCID: 0009-0008-9685-0443) dingpf31@163.com,

Machines(SVM)[11], Particle Filter(PF)[12], and Kernel Adaptive Filter (KAF) algorithms. These methods have strong nonlinear approximation capabilities and have been successfully applied in the prediction of SOH and RUL for Libs. For example, Wang et al.[10]used an algorithm framework combining the self-attention mechanism and LSTM to predict the remaining useful life (RUL) of lithium batteries, using the NASA lithium battery dataset. With a prediction starting point of 80 cycles, the lowest RMSE value achieved was 0.002. AI-Greer M et al.[12] extracted coefficients directly related to the degradation phenomena of lithium batteries' RUL using a reduced-order single-particle model. The resulting data was processed using a smooth particle filter algorithm to obtain the RUL prediction results for the battery. DBN consists of multiple layers of Restricted Boltzmann Machines (RBM) forming a deep neural network. They can fully utilize large-scale, nonlinear, high-dimensional continuous detection data to effectively extract deep features of battery degradation. This allows for the abstraction of high-dimensional raw detection data to lower-dimensional representations, thereby avoiding the inefficiencies and uncertainties of manual feature extraction. DBN receive data through the visible layer and extract data features through hidden layers[13], thereby reducing the original complexity of the data. Liu et al[14].harness the robust feature extraction capabilities of DBNs to examine battery characteristics, utilizing battery current, terminal voltage, and temperature as input parameters.

On the other hand, KAF represents a nonlinear adaptive filtering algorithm based on kernel learning, which extends adaptive filtering algorithms into the Reproducing Kernel Hilbert Space. In KAF, utilizing kernel functions simplifies the original computational process by transforming a low-dimensional nonlinear space into a high-dimensional linear space through the "kernel trick." This transformation provides a distinct advantage in solving nonlinear problems[15-17]. As nonlinear systems composed of time series become increasingly complex, the KAF distinguishes itself with its online prediction capabilities. Unlike methods such as LSTM and SVM, KAF can provide real-time outputs for complex time series predictions, resulting in better performance in time series forecasting.

While time series prediction based on the KAF offers real-time forecasting and tracking of time-varying features, there are still some shortcomings in the process. As new samples are added, the system's memory usage continues to increase, and the computational complexity escalates with the growing volume of samples. These advantages and disadvantages are particularly pronounced in the KRLS algorithm.

To address the issue of increasing computational complexity in the KRLS algorithm, Van et al.[18] introduced the Sliding Window Kernel Recursive Least Squares (SW-KRLS) algorithm. This method involves computing with a fixed-length sequence of samples, thereby reducing the dimensionality of the kernel matrix. Liu et al.[19] introduced the Extended Kernel Recursive Least Squares (EX-KRLS) algorithm, which expresses state space models within the kernel space, thereby enhancing prediction accuracy. Chen et al.[20] introduced the Quantization Kernel Recursive Least Squares (QKRLS) algorithm, which compresses input data through quantization operations, thereby limiting the matrix dimensions and enhancing prediction accuracy. Van et al.[21] introduced the FB-KRLS algorithm, which, similar to the SW-KRLS method, computes using a fixed-length sequence of samples. However, FB-KRLS differs in its selection process by retaining data samples with stronger relevance and discarding those with weaker relevance, thereby reducing data dimensions and improving prediction accuracy.

Drawing from the aforementioned literature, this paper presents an FB-KRLS algorithm rooted in the SCKF method, integrated with DBN for extracting features from raw data to predict the RUL of lithium batteries. FB-KRLS reduces the computational complexity of the KRLS algorithm by implementing a Fixed-Budget (FB) criterion. Compared to other kernel learning methods, the KRLS algorithm uses a recursive approach to update parameters, enabling it to process large batches of data online. This makes it particularly suitable for time series prediction. Integrating the SCKF method enhances the effectiveness of the FB-KRLS algorithm, allowing it to more accurately mimic the actual behavior of nonlinear systems. Moreover, employing a DBN for feature extraction on the dataset reduces its complexity, thereby improving the overall predictive accuracy of the algorithm. Under identical conditions, this study compares the proposed algorithm with various methods including LSTM, SVR, KLMS, KRLS, EX-KRLS, SW-KRLS, Q-KRLS, FB-KRLS, SCKF-KRLS, CKF-FB-KRLS and SCKF-FB-KRLS to validate the effectiveness of

the algorithm.

2. Methodology

2.1. DBN

DBN are a type of semi-supervised learning method based on RBMs. RBM is an unsupervised learning algorithm based on artificial neural networks. The term "restricted" in RBM refers to the model structure where layers are fully connected between each other, yet there are no intra-layer connections. The RBM comprises two layers: a visible layer that inputs data and a hidden layer that learns the latent features of the data[22,23]. The network structure is depicted in Fig. 1.

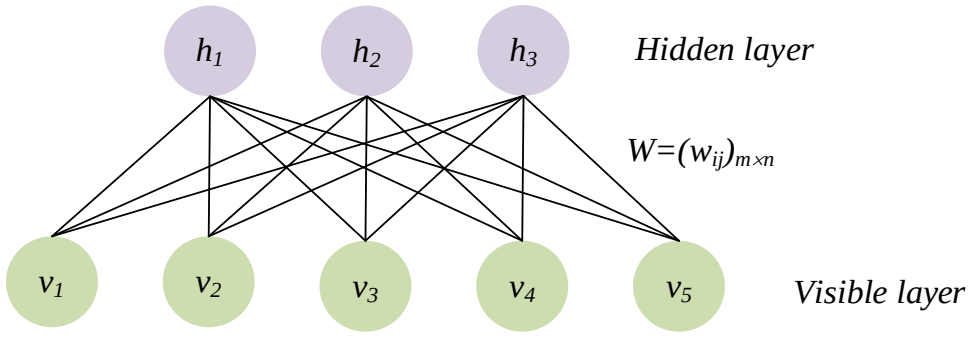


Fig. 1. Structure of RBM model.

In this model,$V = (v_1, v_2, \cdots v_m)^T$ denotes the state vector of the visible layer, $H = (h_1, h_2, \cdots h_n)^T$ denotes the state vector of the hidden layer, m denotes the number of neurons in the visible layer, n denotes the number of neurons in the hidden layer, and $W=(w_{ij})_{m\times n}$ is the weight matrix connecting the visible and hidden layers. Given training data, V and H are transformed into each other through W.

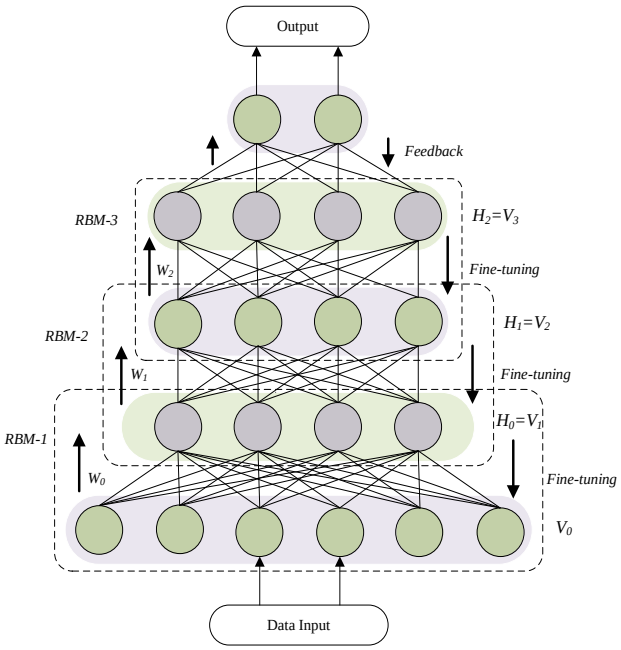


Fig. 2. Structure of DBN model

A complete DBN model is composed of two or more stacked RBM, where the quantity of neurons in each tier is dictated by the particular issue. The learning procedure of the framework includes two segments: unsupervised preliminary training and supervised fine-tuning[24,25]. The pre-training utilizes a greedy layer-by-layer unsupervised learning of numerous RBM layers, whereas the fine-tuning entails modifying the model parameters using the backpropagation algorithm. The detailed network configuration is illustrated Fig. 2.

As depicted in the diagram, the initial layer in Fig. 2. is the visible layer for the input data, and the neuron count corresponds to the dimensions of the input data. The visible layer V_0 and the hidden layer H_0 together form RBM-1, which starts the layer-by-layer training to extract data features. The hidden layer of RBM-1 functions as the input layer for RBM-2, and this process continues in a similar fashion to complete the parameter initialization for the pretraining phase[26].

2.2. FB-KRLS

The KRLS model was first introduced by Engle et al[17], applying kernel methods to the Recursive Least Squares (RLS) algorithm for nonlinear applications. In this process, low-dimensional inputs are mapped into a high-dimensional space via a kernel function, which can be expressed as:

$$\boldsymbol{\phi}: \boldsymbol{X} \to \boldsymbol{F}, \boldsymbol{x} \to \boldsymbol{\phi}(\boldsymbol{x}) \tag{1}$$

In this model, $\boldsymbol{X}$ represents the original space, that is, the low-dimensional space, denoted as$\boldsymbol{X} = \{x_t\} \in \mathbb{R}^N$. $\boldsymbol{F}$ denotes the mapped space, denoted as $\boldsymbol{F} = \{\boldsymbol{\phi}(\boldsymbol{x})\} \in \mathbb{R}^M$. Since $\boldsymbol{\phi}$ is difficult to determine, a kernel function is used to assist in this process. Typically, the kernel function is chosen to be the Gaussian kernel, denoted as $\kappa(\boldsymbol{x_i}, \boldsymbol{x_t}) = \langle \varphi(u_i), \varphi(u_t) \rangle$. Here,

$\langle \varphi(u_i), \varphi(u_t) \rangle = \langle \boldsymbol{u_i}, \boldsymbol{u_t} \rangle$ is referred to as the inner product.

During the training phase, after obtaining n input-output pairs sequentially, the goal of the KRLS algorithm is to determine the optimal coefficient vector α through the least squares method, such that the loss function in Eq.(2) is minimized[21], i.e.

$$J(\alpha) = \min_{\alpha} |y - \boldsymbol{K}\alpha|^2 + \lambda \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} \tag{2}$$

In this context, y, $\boldsymbol{K}$, and λ represent the vector containing outputsy_i, the kernel matrix, and the regularization parameter, respectively, where the components of the kernel matrix are $\boldsymbol{K}_{i,j} = k(\boldsymbol{s_i}, \boldsymbol{s_j})$. The solution to Eq.(2) is:

$$\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} \boldsymbol{y} \tag{3}$$

$\boldsymbol{I}$ represents the identity matrix.

Due to the KRLS algorithm, there is an issue of unbounded growth of the kernel matrix $\boldsymbol{K}$ during the training phase as the number of input data continuously increases. To tackle this issue, researchers have suggested approaches such as the ALD criterion and "sliding window" techniques. The essence of these methods is to create a "dictionary" by extracting data that meets predetermined conditions, thereby reducing the dimensions of the kernel matrix and limiting its unbounded growth. The FB-KRLS algorithm employs a fixed budget memory method, known as the FB criterion. This criterion only comes into effect when the dictionary's size surpasses a preset threshold M. As mentioned in the literature[18], if the number of samples that meet the FB criterion is greater than M, the kernel matrix is managed according to Eq.(4) When a new sample meets the required conditions, less important data are discarded from the dictionary as outlined in Eq.(5) Conversely, if a new sample does not meet these conditions, it is not added to the dictionary, thus keeping the size of the dictionary unchanged. As new qualifying data are added, the least important data are removed to maintain the dictionary size. As mentioned in the literature[21], to keep the dimensions of the kernel matrix constant, an error discarding criterion is considered as shown in Eq.(6).

$$\breve{\boldsymbol{K}}_t^{-1} = \begin{bmatrix} \boldsymbol{K}_{t-1}^{-1} + gee^T & -ge \\ -ge^T & g \end{bmatrix} \tag{4}$$

$$\breve{\boldsymbol{K}}_t^L = \begin{bmatrix} a & k_{t-1}(u_t)^T \\ k_{t-1}(u_t) & \boldsymbol{K}_t^L \end{bmatrix}, (\breve{\boldsymbol{K}}_t^L)^{-1} = \begin{bmatrix} e & \boldsymbol{f}^T \\ \boldsymbol{f} & \boldsymbol{G} \end{bmatrix}$$

$$\Rightarrow (\boldsymbol{K}_t^L)^{-1} = \boldsymbol{G} - \boldsymbol{f}\boldsymbol{f}^T / e \tag{5}$$

$$d_L(\boldsymbol{u_i}, y_i) = \min_i \left(\frac{|\alpha_i|}{[\breve{K}_t^{-1}]_{i,i}} \right) \tag{6}$$

In this context, L represents the index of the least significant data in the memory dictionary. The term $\breve{\boldsymbol{K}}_t^L$ refers to the kernel matrix after new rows and columns have been added. The variables $e = \boldsymbol{K}_{t-1}^{-1} b$, $g = (d - b^T e)^{-1}$, $b^T f + dg = 1$, $\boldsymbol{K}_t^L$ indicate the kernel matrix after the L-th row and L-th column have been excised. $\boldsymbol{\alpha}_i$ represents the i-th element in matrix $\boldsymbol{\alpha}$.

2.3. SCKF-FB-KRLS

The FB-KRLS algorithm based on SCKF adopts the concept of dual alternating learning, and its structural diagram is illustrated in Fig. 3. From the diagram, it is evident that the entire algorithm comprises two sections: the FB-KRLS algorithm and the SCKF algorithm. The input data for the model is "(u_t, y_t) ". During the entire algorithm training, SCKF provides the FB-KRLS algorithm with the necessary inputs for learning and training. In return, FB-KRLS provides SCKF with the measurement outputs required for state estimation. The two algorithms influence each other, and as the number of training iterations increases, the FB-KRLS algorithm refines the measurement equation of the nonlinear system with the help of SCKF, thereby making the overall algorithm more in line with the formal situation of the nonlinear system.

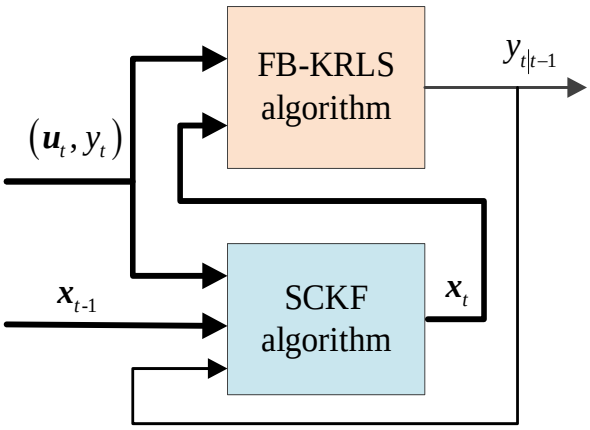


Fig. 3. Overall Implementation Framework of the FB-KRLS Method Based on SCKF

Due to the need for square root calculations of the state covariance matrix in the Cubature Kalman Filter (CKF) during computation, if certain conditions are met, it can lead to interruptions in filtering, affecting the stability of the algorithm. Therefore, a square root variant of the Cubature Kalman Filter algorithm has been suggested[27]. The SCKF algorithm is an approximate Kalman filtering method established in the Gaussian domain, which calculates the statistics of random

variables after nonlinear transformation by generating cubature points[28]. The SCKF algorithm solves the problem of numerical stability and reduces computational load, thus offering better performance. This paper integrates the SCKF algorithm with the FB-KRLS algorithm to construct the SCKF-FB-KRLS algorithm. Here are the detailed steps of the algorithm:

1) Initialize the regularization parameter λ, the noise covariance matrices $\boldsymbol{Q}_1, R_1, \boldsymbol{x_1}$ of the inputs, the square root factor of $\boldsymbol{P}_1$ is $\boldsymbol{S}_1$, where $\boldsymbol{P}_1$ is the filtering state covariance matrix.

2) When $t = 1$, initialize the measurement equation of the KRLS algorithm to obtain $\alpha_1 = \boldsymbol{K}_1^{-1} y_1$ and $\boldsymbol{K}_1 = [\lambda + k(\boldsymbol{u_1}, \boldsymbol{u_1})]$.

3）Calculate the cubature points $\widehat{\boldsymbol{X}}_{t-1}^{(i)}$.

$$\widehat{\boldsymbol{X}}_{t-1}^{(i)} = \boldsymbol{x}_{t-1} + \boldsymbol{S}_{t-1}\xi^{(i)}, \quad i=1,\dots,2n, \quad t = 2,3,4\dots, \tag{7}$$

Here, $\xi^{(i)}$ is the weight matrix for the cubature points, $\xi^{(i)} = \begin{cases} \sqrt{n}e_i, & i = 1,2,\dots,n \\ -\sqrt{n}e_{i-n}, & i = n+1, n+2, \dots, 2n \end{cases}$, $e_i \in R^n$ are the unit vectors in the positive direction along the coordinate axes.

4) Calculate the propagation of cubature points.

$$\widehat{\boldsymbol{X}}_{t|t-1}^{(i)} = f\left(\widehat{\boldsymbol{X}}_{t-1}^{(i)}\right), \quad i=1,\dots,2n \tag{8}$$

5) Compute the predicted state $\boldsymbol{x}_{t|t-1}$:

$$\boldsymbol{x}_{t|t-1} = \frac{1}{2n}\sum_{i=1}^{2n} \widehat{\boldsymbol{X}}_{t|t-1}^{(i)} \tag{9}$$

6) Compute the square root factor $\boldsymbol{S}_{t|t-1}$ of the predicted error covariance:

$$\boldsymbol{S}_{t|t-1} = Tria\left(\left[\widehat{\boldsymbol{\chi}}_{t|t-1} \quad \boldsymbol{S}_{Q,t-1}\right]\right) \tag{10}$$

$$\widehat{\boldsymbol{\chi}}_{t|t-1} = \frac{1}{\sqrt{2n}}\left[\widehat{\boldsymbol{X}}_{t|t-1}^{(1)} - \boldsymbol{x}_{t|t-1} \quad \widehat{\boldsymbol{X}}_{t|t-1}^{(2)} - \boldsymbol{x}_{t|t-1} \quad \dots \quad \widehat{\boldsymbol{X}}_{t|t-1}^{(2n)} - \boldsymbol{x}_{t|t-1}\right] \tag{11}$$

Here, $\widehat{\boldsymbol{\chi}}_{t|t-1}$ denotes the weighted mean matrix, $\boldsymbol{S}_{Q,t-1}$ is the square root factor of the process noise covariance matrix, and $Tria(\cdot)$ signifies the triangularization of the matrix.

7) Update the cubature points of the predicted state vector $\boldsymbol{X}_{i,t|t-1}$:

$$\mathbf{X}_{t|t-1}^{(i)} = \mathbf{x}_{t|t-1} + \mathbf{S}_{t|t-1}\xi^{(i)}, \quad i=1,\dots,2n \tag{12}$$

8) Through the measurement equation, determine the propagation of the cubature points calculated by the FB-KRLS algorithm:

$$\boldsymbol{Y}_{t|t-1}^{(i)} = \boldsymbol{k}_{t-1}^T\left(\boldsymbol{X}_{t|t-1}^{(i)}\right)\boldsymbol{\alpha}_{t-1}, \quad i = 0,\dots,2n \tag{13}$$

9) Obtain the output matrix $y_{t|t-1}$, the cross-covariance matrix $\boldsymbol{P}_{xy,t|t-1}$, and the predicted covariance matrix $\boldsymbol{S}_{yy,t|t-1}$.

$$y_{t|t-1} = \frac{1}{2n}\sum_{i=1}^{2n} \boldsymbol{Y}_{t|t-1}^{(i)} \tag{14}$$

$$\boldsymbol{S}_{yy,t|t-1} = Tria\left(\left[\boldsymbol{\gamma}_{t|t-1} \quad \boldsymbol{S}_{R,t}\right]\right) \tag{15}$$

$$\boldsymbol{P}_{xy,t|t-1} = \boldsymbol{\chi}_{t|t-1}\boldsymbol{\gamma}_{t|t-1}^T \tag{16}$$

Here, $\boldsymbol{\gamma}_{t|t-1}^T$ and $\boldsymbol{\chi}_{t|t-1}$ are both weighted mean matrices, and $\boldsymbol{S}_{R,t}$ is the square root factor of the process noise covariance matrix $\boldsymbol{R}_t$.

$$\boldsymbol{\gamma}_{t|t-1} = \frac{1}{\sqrt{2n}}\left[\boldsymbol{Y}_{t|t-1}^{(1)} - \boldsymbol{y}_{t|t-1} \quad \boldsymbol{Y}_{t|t-1}^{(2)} - \boldsymbol{y}_{t|t-1} \quad \dots \quad \boldsymbol{Y}_{t|t-1}^{(2n)} - \boldsymbol{y}_{t|t-1}\right] \tag{17}$$

$$\boldsymbol{\chi}_{t|t-1} = \frac{1}{\sqrt{2n}}\left[\boldsymbol{X}_{t|t-1}^{(1)} - \boldsymbol{x}_{t|t-1} \quad \boldsymbol{X}_{t|t-1}^{(2)} - \boldsymbol{x}_{t|t-1} \quad \dots \quad \boldsymbol{X}_{t|t-1}^{(2n)} - \boldsymbol{x}_{t|t-1}\right] \tag{18}$$

10) Compute the Kalman gain $\boldsymbol{G}_t$, and the predicted error covariance matrix $\boldsymbol{S}_t$, update the state matrix $\boldsymbol{x}_t$.

$$\boldsymbol{G}_t = \boldsymbol{P}_{xy,t|t-1}\boldsymbol{P}_{yy,t|t-1}^{-1} \tag{19}$$

$$\boldsymbol{x}_t = \boldsymbol{x}_{t|t-1} + \boldsymbol{G}_t\left(y_t - y_{t|t-1}\right) \tag{20}$$

$$\boldsymbol{S}_t = Tria\left(\left[\boldsymbol{\chi}_{t|t-1} - \boldsymbol{G}_t\boldsymbol{\gamma}_{t|t-1} \quad \boldsymbol{G}_t\boldsymbol{S}_{R,t}\right]\right) \tag{21}$$

11) Construct the input vector $\boldsymbol{z}_t = [\boldsymbol{x}_t^T \ \boldsymbol{u}_t^T]^T$ and apply the FB-KRLS algorithm, with updates calculated using $\boldsymbol{k}_{t-1}(\boldsymbol{z}_t) = [k(\boldsymbol{z}_1, \boldsymbol{z}_t), k(\boldsymbol{z}_2, \boldsymbol{z}_t), \dots, k(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t)]^T$.

$$\boldsymbol{\alpha}_t = \boldsymbol{K}_t^{-1}\boldsymbol{y}_t \tag{22}$$

12) Return to step 3) and repeat until the training is complete.

2.4. RUL framework predictions

Based on the algorithm mentioned above, the proposed framework for predicting the RUL of lithium-ion batteries is illustrated Fig. 4. This framework primarily consists of the following four parts:

(1) Data Pre-processing. Features including discharge current, discharge voltage, and battery temperature, discharge time, which reflect the degradation trend of the RUL, are selected and subjected to normalization.

(2) Feature Extraction. Utilize a 4-layer RBM structure of a Deep Belief Network to perform deep feature extraction on the existing dataset, obtaining deep features of lithium battery data and completing the unsupervised learning process.

(3) Model Training. Set the prediction start point and use it as a delimiter to partition the dataset into training and testing sets. Use the deep features extracted by the optimized DBN as inputs and the corresponding capacity as outputs to construct the SCKF-FB-KRLS model.

(4) RUL Prediction. Input the test dataset into the SCKF-FB-KRLS model to obtain the predicted capacity for each cycle. Finally, calculate the RUL of the lithium battery using the termination threshold.

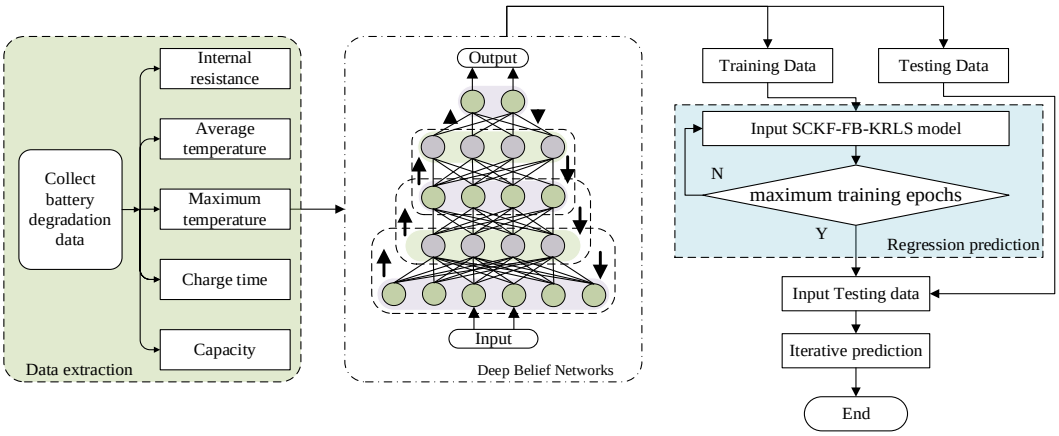


Fig. 4. Structural Diagram for Lithium Battery RUL Prediction.

3. Experimental results and analysis

3.1. Introduction to lithium battery datasets

The dataset utilized in this article is an open battery dataset supplied by the Prognostics Center of Excellence at the NASA. The dataset includes experiments on 18650-type lithium cobalt oxide batteries, specifically models B0005, B0006, B0007, and B0018, conducted at an environmental temperature of 24°C. The charge and discharge experimental procedures for these batteries are as follows:

(1) Charging: Charge in constant current mode at 1.5A until the battery voltage attains 4.2V, then switch to constant voltage charging until the charging current drops to 20mA.

(2) Discharging: Discharge in constant current mode at 2A until the voltage of B0005, B0006, B0007, and B0018 decreases to 2.7V, 2.5V, 2.2V, and 2.5V respectively.

Table 1. displays the parameters of the battery throughout the charging and discharging cycles, which include environmental temperature (ET), charging current (CC), discharging current (DC), end of discharge voltage (EOC), and end of life criterion (EOLC). The lifespan of the battery is deemed complete once its capacity declines to 70% of its nominal capacity. Fig. 5. illustrates the capacity degradation curves for batteries B0005, B0006, B0007, and B0018. It is observed that the battery capacity tends to diminish with an increasing number of discharge cycles. Nonetheless, some upward fluctuations are observed during this process, attributed to the phenomenon of internal capacity regeneration in the batteries. The first three sets of batteries each have data for 168 cycle groups, while B0018 has data for only 132 cycle groups.

Table 1. Battery Charge and Discharge Parameters.

No.	ET/℃	CC/A	DC/A	EOC/V	EOLC/%
B0005	24	1.5	2	2.7	30
B0006				2.5	
B0007				2.2	
B0018				2.5	

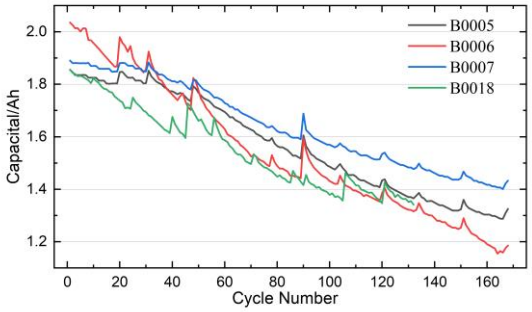


Fig. 5. Battery capacity degradation curve.

3.2. Evaluation criteria for prediction accuracy of lithium battery RUL

This paper adopts several evaluation metrics for RUL prediction issues, such as Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE)and Relative Error (RE) to examine the predictive outcomes. As mentioned in in the literature[12], the equations are presented below:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \times 100\% \tag{23}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \tag{24}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{25}$$

$$RE = 1-\frac{|RUL^{pred}-RUL^{ture}|}{RUL^{ture}} \times 100\% \tag{26}$$

In the above formulas, n represents the total number of samples in the prediction dataset,y_i indicates the actual battery capacity, and $\hat{y}_i$ refers to the estimated outcome from the prediction model.$RUL^{pred}$and RUL^{ture} represent the actual and predicted values of the remaining life, respectively. For the evaluation metrics MAPE, RMSE, and MAE, lower values indicate superior predictive performance. In contrast, a value of RE closer to 1 signifies enhanced prediction accuracy.

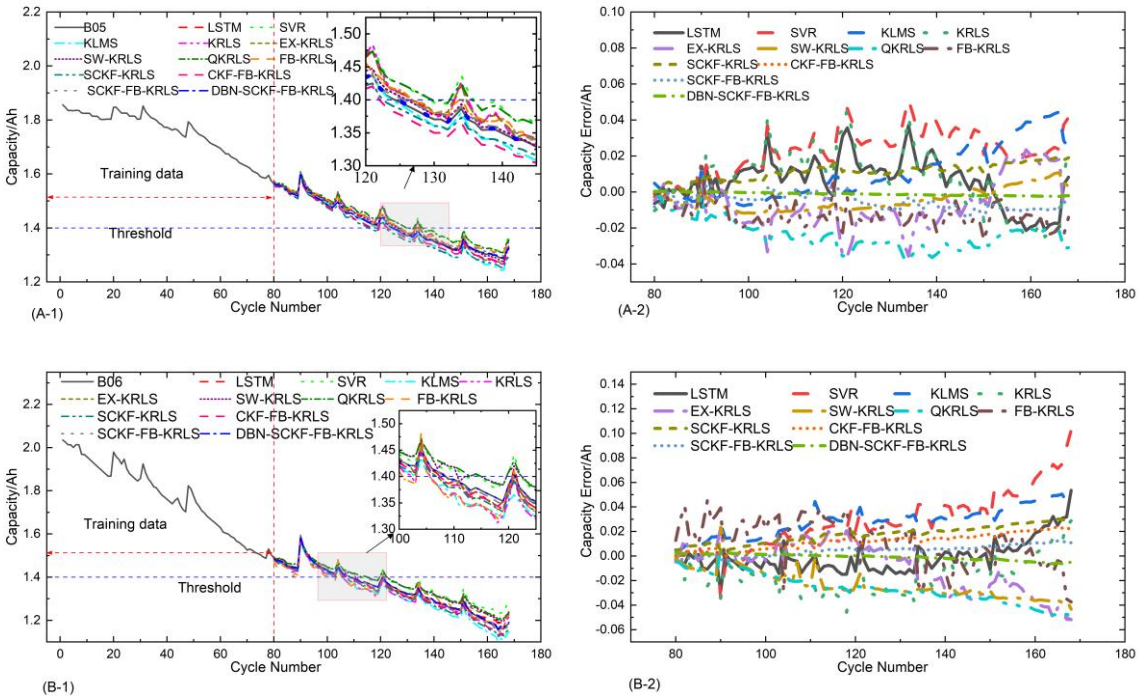
3.3. Comparative experiment on the prediction effect of lithium battery RUL

Using the experimental dataset mentioned, predictive models are trained using various algorithms including LSTM, SVR, KLMS, KRLS, EX-KRLS, SW-KRLS, Q-KRLS, FB-KRLS, SCKF-FB-KRLS, and DBN-SCKF-FB-KRLS, The effectiveness of each algorithm is then compared. The configuration details for many of the aforementioned algorithms are outlined below: in the LSTM algorithm, the parameters are set as follows: the LSTM layer has 100 neurons, and the fully connected layer has 15 neurons. For the SVR algorithm, the kernel function is set to Radial Basis Function (RBF), The coefficient parameter γ for the RBF kernel is set to 1, while the penalty coefficient is set to C=10, and the tolerance parameter epsilon=0.001. For the subsequent KRLS algorithm and its variants, the RBF is selected as the kernel function. The RBF kernel parameter is $\sigma = 3$, and the regularization parameter is $\lambda = 10^{-3}$. For the EX-KRLS algorithm, the settings are as follows: the scale factor a=0.995, the forgetting factor $\beta = 0.9$, and the noise factor $q = 1e^{-4}$. In the FB-KRLS algorithm, the learning rate $\eta = 0.1$. For both the SW-KRLS and FB-KRLS algorithms, the dictionary size is established at M = 200. Q-KRLS algorithm quantization threshold $\varepsilon_F = \varepsilon_U = 0.1$. In the SCKF algorithm, Initial state covariance matrix $\boldsymbol{P} = 0.09\boldsymbol{I}$,$\boldsymbol{I}$ is the identity matrix, Noise covariance matrix $\boldsymbol{Q} = 0.01\boldsymbol{I}, R = 0.01$. The DBN used in here is composed of three layers of RBM.

The RUL prediction model for lithium batteries, employing the DBN-SCKF-FB-KRLS method, is detailed in Eq.(27):

$$y(t+1) = f\left(x_1(t), x_2(t), \cdots, x_7(t), x_8(t), y(t-1), y(t)\right) \quad (27)$$

In the model, $x_1(t), x_2(t), \cdots, x_7(t), x_8(t)$ represents the feature input of the lithium battery processed by the DBN, and $y(t-1), y(t)$ represents the outputs at times $t-1$ and $t$during the prediction process.



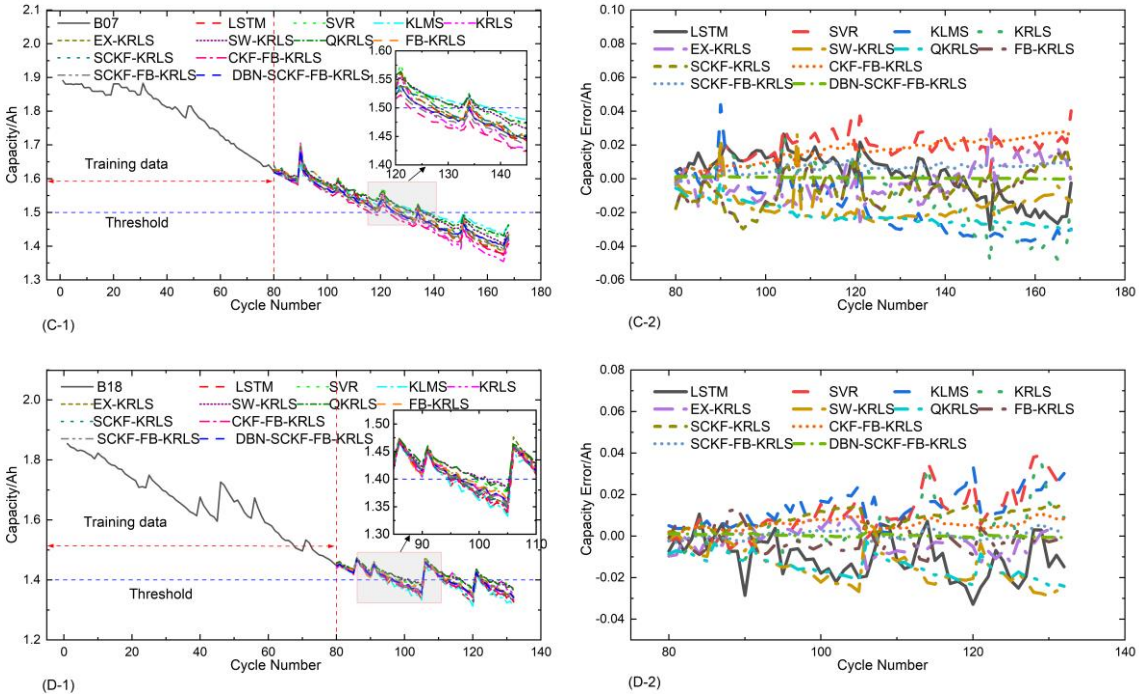
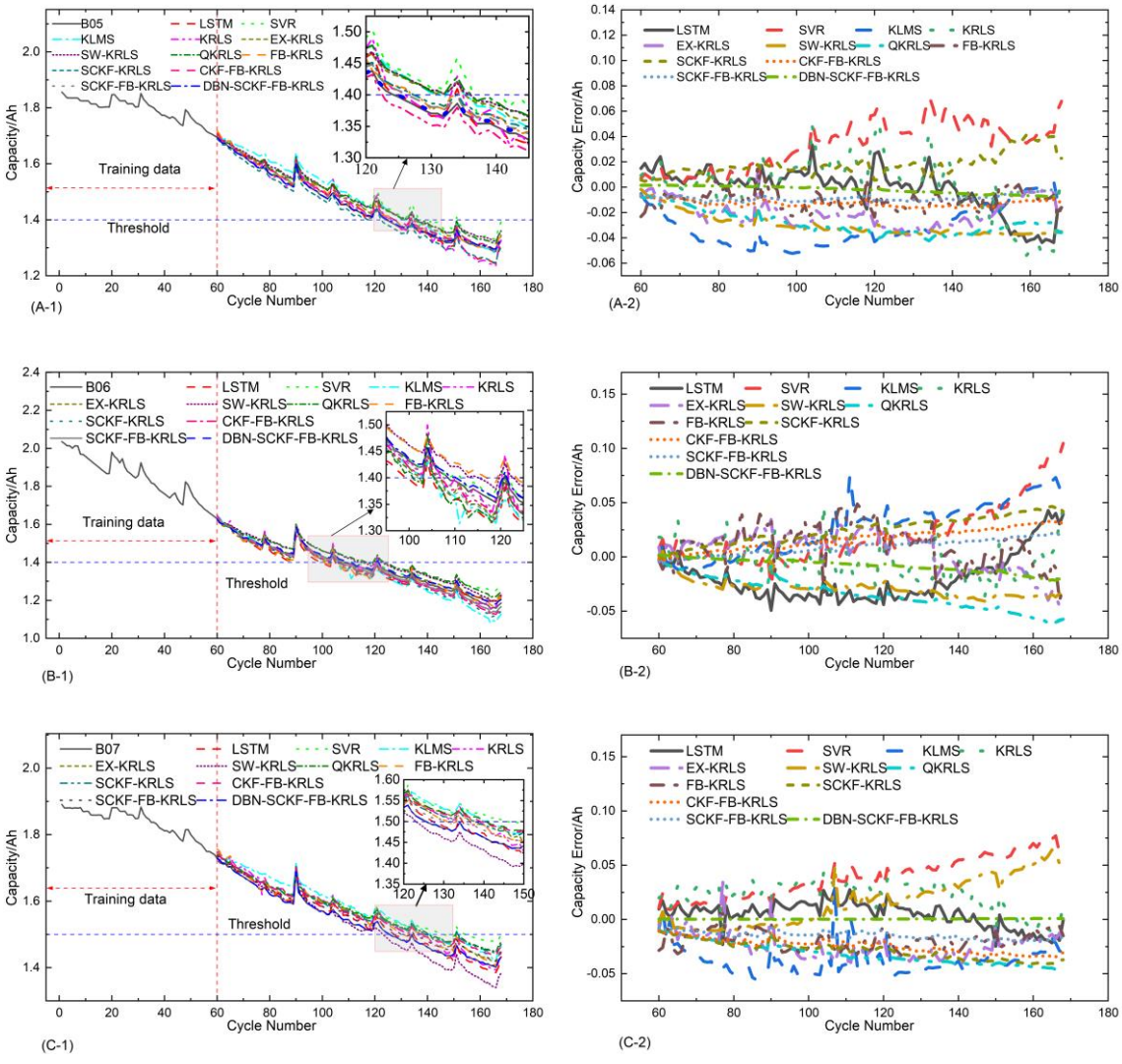


Fig. 6. Lithium battery prediction results based on the above algorithm (80 cycles).



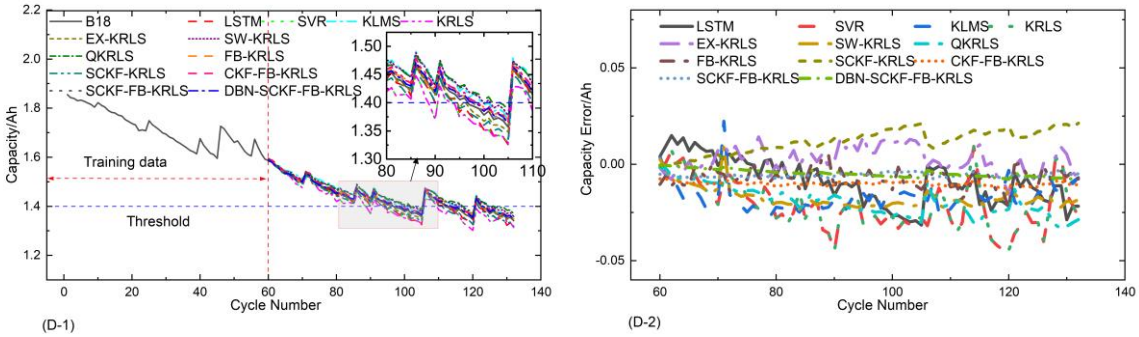


Fig. 7. Lithium battery prediction results based on the above algorithm (60 cycles).

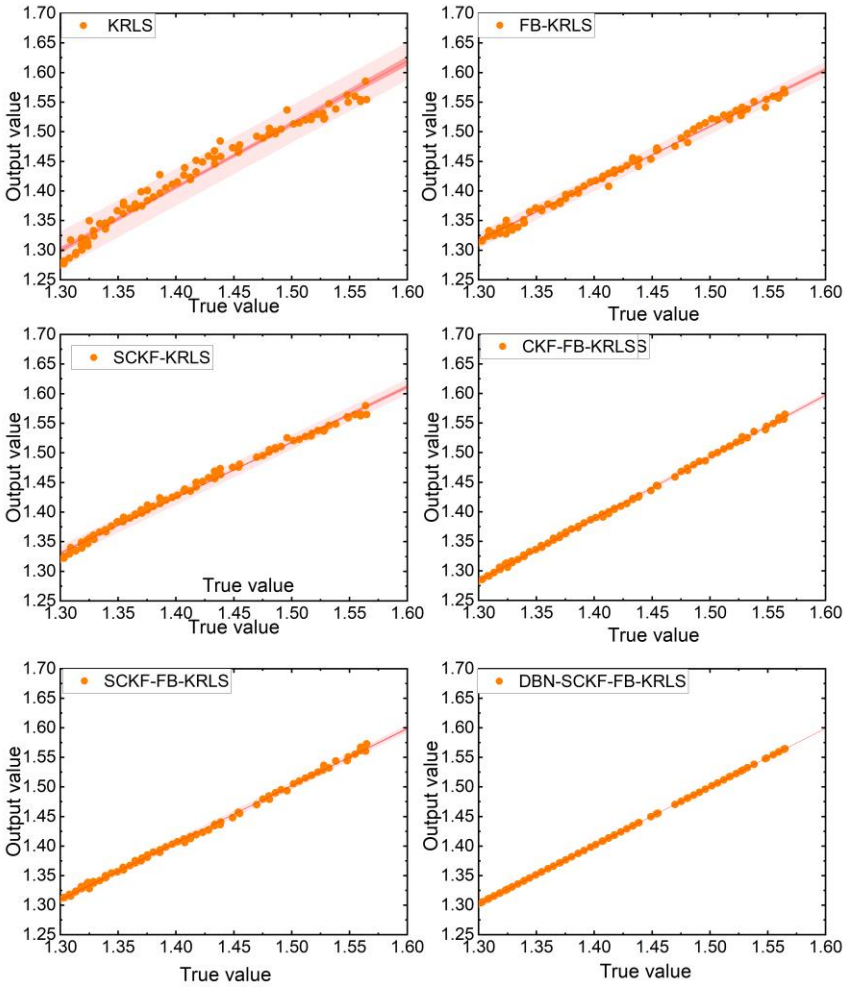


Fig. 8. The fit between the predicted and actual values of battery B0005. (80 Cycle).

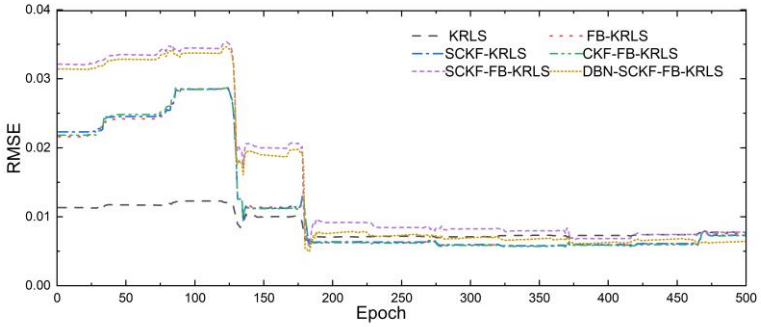


Fig. 9. Error convergence curve of the algorithm.

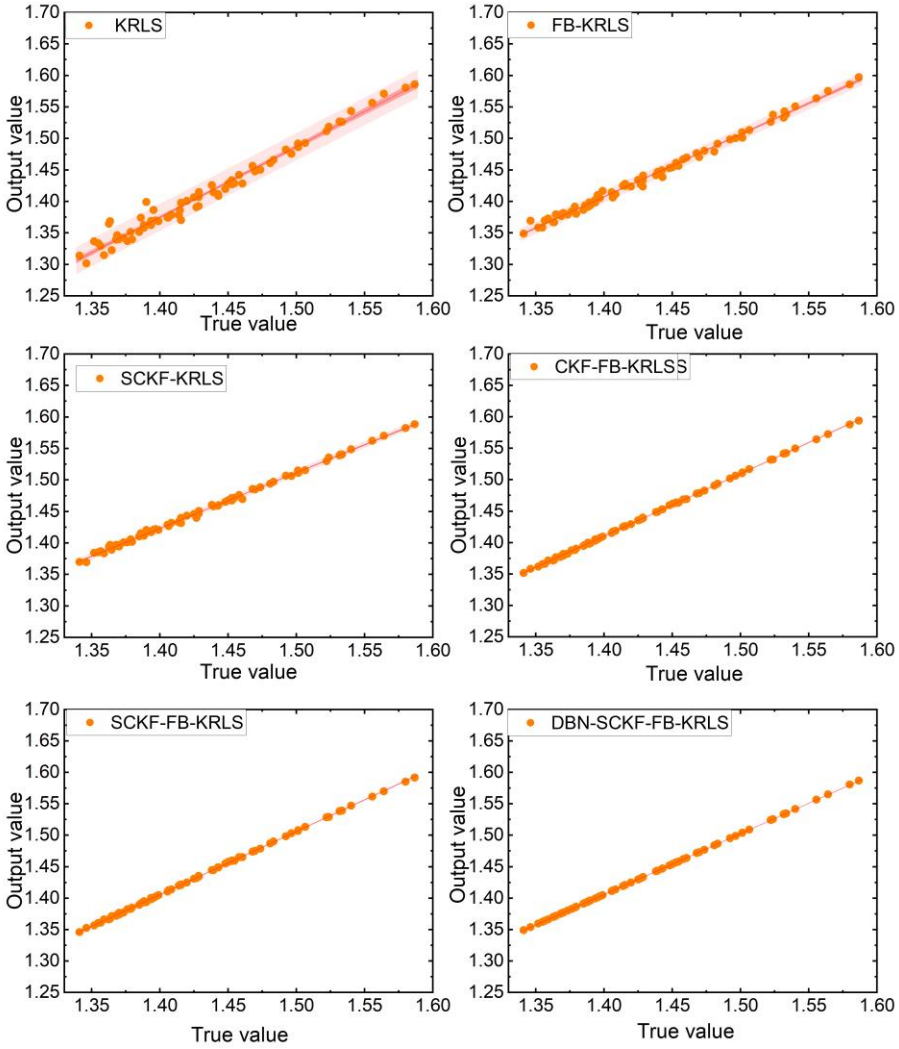


Fig. 10. The fit between the predicted and actual values of battery B0018. (60 Cycle).

In the NASA lithium battery dataset, except for battery B0018, which has a cycle life of over 130 cycles, the rest of the batteries have cycle lives exceeding 160 cycles. Moreover, the end of life for all batteries occurs after at least 80 cycles. Consequently, this study sets cycle 80 as the starting point for the prediction dataset to experiment with forecasting the remaining life of lithium batteries. Subsequently, to compare and validate the prediction effects of different algorithms under longer prediction cycles, cycle 60 is set as the new starting point for the prediction dataset in further experiments.

With the prediction starting point is 80 cycles, the resulting RUL prediction outcomes are shown in Fig. 6. In the graph, the black curve depicts the actual capacity values of the lithium battery. The red dashed vertical line indicates the starting point of the prediction data (80 cycles), and the blue dashed horizontal line signifies that the battery has reached 70% of its capacity (1.4 Ah), marking the end of the battery's life. Given that battery B0007's capacity consistently exceeds 70% (1.4 Ah), the blue dashed line is elevated to the 1.5 Ah mark to enhance visibility of the prediction performance of the discussed models on the lithium battery's RUL.

Observations from Fig. 6. (A-1),(B-1),(C-1),and(D-1) indicate that the discussed algorithms are somewhat effective in demonstrating the degradation trend of lithium battery capacity and in accurately detecting the capacity regeneration phenomenon. This evidence suggests that the algorithms analyzed in this study are effective. Close examination of the enlarged images in Fig. 6. clearly shows that the RUL prediction capabilities of the FB-KRLS algorithm are substantially enhanced with the integration of SCKF and DBN. Compared to other algorithms, the model trained using the DBN-SCKF-FB-KRLS algorithm yields predictions that are closer to the actual values. Observing Fig. 6. (A-2),(B-2),(C-2),and(D-2), the prediction errors for lithium battery capacity fluctuate between

0.10 Ah and -0.06 Ah. Specifically, the prediction error for battery B0018 ranges from -0.04 Ah to 0.04 Ah. Compared to the LSTM and SVR methods, the error fluctuation is smaller with the kernel adaptive filtering techniques. Both the SCKF-FB-KRLS and DBN-SCKF-FB-KRLS algorithms show a significant decrease in prediction errors for lithium battery capacity, with the error of the DBN-SCKF-FB-KRLS algorithm approaching zero. This underscores the appropriateness of this method for forecasting the RUL of lithium batteries.

Adjusting the initial prediction starting point from 80 to 60 cycles extends the model's forecasting period, with the resulting RUL predictions are displayed in Fig. 7. (A-1),(B-1),(C-1)and(D-1). The comparison shows that under longer prediction cycles, the discrepancy between the predictions of various algorithms and the actual values tends to increase. However, the model developed with the DBN-SCKF-FB-KRLS algorithm produces predictions that align more closely with actual values, outperforming those generated by other training models. This highlights the effectiveness of the DBN-SCKF-FB-KRLS approach in achieving more accurate RUL predictions. Analysis of Fig. 7. (A-2),(B-2),(C-2)and(D-2) reveals that the prediction error for lithium battery capacity oscillates between -0.15Ah and 0.25Ah, with the specific error for battery B0018 ranging from -0.04Ah to 0.04Ah. This results from the fewer charging and discharging cycles of B0018, leading to a smaller dataset. Consequently, the training set selection comprises a larger proportion of the total dataset.

Table 2. Prediction results of different models for B0005 (80 Cycle).

No.	Method	RMSE	MAE	MAPE	RE
B0005	LSTM[10]	0.0608	0.0348	2.530%	0.9556
	SVR[11]	0.0470	0.0254	1.840%	0.8889
	KLMS[16]	0.0319	0.0252	1.827%	0.9778
	KRLS[17]	0.0484	0.0263	1.900%	0.9556
	EX-KRLS[19]	0.0410	0.0257	1.841%	0.9556
	SW-KRLS[18]	0.0408	0.0239	1.750%	0.9556
	QKRLS[20]	0.0297	0.0237	1.702%	0.8889
	FB-KRLS[21]	0.0288	0.0244	1.747%	0.9556
	SCKF-KRLS	0.0233	0.0202	1.303%	0.9333
	CKF-FB-KRLS	0.0157	0.0136	0.984%	0.9556
	SCKF-FB-KRLS	0.0108	0.0071	0.509%	0.9778
	DBN-SCKF-FB-KRLS	0.0022	0.0037	0.262%	1.0000

Table 3. Prediction results of different models for B0006 (80 Cycle)

No.	Method	RMSE	MAE	MAPE	RE
B0006	LSTM[10]	0.0472	0.0350	2.650%	0.9655
	SVR[11]	0.0451	0.0296	2.300%	0.8966
	KLMS[16]	0.0414	0.0322	2.448%	0.8966
	KRLS[17]	0.0400	0.0282	2.090%	0.8966
	EX-KRLS[19]	0.0389	0.0258	1.961%	0.9310
	SW-KRLS[18]	0.0370	0.0245	1.868%	0.9310
	QKRLS[20]	0.0291	0.0257	1.970%	0.8621
	FB-KRLS[21]	0.0285	0.0231	1.694%	0.8966
	SCKF-KRLS	0.0205	0.0179	1.365%	0.9655
	CKF-FB-KRLS	0.0157	0.0131	1.007%	0.9655
	SCKF-FB-KRLS	0.0089	0.0072	0.548%	1.0000
	DBN-SCKF-FB-KRLS	0.0016	0.0024	0.186%	1.0000

Table 4.Prediction results of different models for B0007 (80 Cycle).

No.	Method	RMSE	MAE	MAPE	RE
B0007	LSTM[10]	0.0489	0.0296	1.980%	0.8977
	SVR[11]	0.0379	0.0231	1.550%	--
	KLMS[16]	0.0402	0.0258	1.731%	--
	KRLS[17]	0.0447	0.0272	1.830%	0.8523
	EX-KRLS[19]	0.0309	0.0201	1.334%	0.9432
	SW-KRLS[18]	0.0413	0.0204	1.355%	--
	QKRLS[20]	0.0433	0.0214	1.434%	--
	FB-KRLS[21]	0.0233	0.0181	1.191%	0.9545

	SCKF-KRLS	0.0246	0.0199	1.205%	0.8977
	CKF-FB-KRLS	0.0129	0.0110	0.739%	0.9318
	SCKF-FB-KRLS	0.0118	0.0100	0.667%	0.9545
	DBN-SCKF-FB-KRLS	0.0018	0.0021	0.139%	--

Fig. 8. shows the fitting effects of the model predictions compared to the actual values for battery B0005, starting from the 80th cycle. It is evident that among these four models, the fitting effect of SCKF-FB-KRLS is superior to that of KRLS and FB-KRLS, while DBN-SCKF-FB-KRLS shows an even better fitting performance than SCKF-FB-KRLS. Therefore, the DBN-SCKF-FB-KRLS algorithm has certain advantages in nonlinear prediction. Similarly, Fig. 10. shows the fitting effects of the four algorithms compared to the actual values for battery B0018, starting from the 60th cycle. The results are similar to those presented inFig. 8. .Fig. 9. shows the iterative error convergence curves for the model

.

Table 5. Prediction results of different models for B0018 (80 Cycle).

No.	Method	RMSE	MAE	MAPE	RE
B0018	LSTM[10]	0.0469	0.0354	2.530%	0.8824
	SVR[11]	0.0299	0.0218	1.570%	0.9412
	KLMS[16]	0.0421	0.0235	1.684%	0.8824
	KRLS[17]	0.0307	0.0213	1.530%	0.8824
	EX-KRLS[19]	0.0168	0.0165	1.173%	1.0000
	SW-KRLS[18]	0.0198	0.0150	1.084%	0.7059
	QKRLS[20]	0.0289	0.0147	1.054%	0.8235
	FB-KRLS[21]	0.0202	0.0148	1.054%	1.0000
	SCKF-KRLS	0.0132	0.0113	0.813%	1.0000
	CKF-FB-KRLS	0.0104	0.0089	0.637%	1.0000
	SCKF-FB-KRLS	0.0084	0.0065	0.463%	1.0000
	DBN-SCKF-FB-KRLS	0.0019	0.0014	0.102%	1.0000

Table 6. Prediction results of different models for B0005 (60 Cycle).

No.	Method	RMSE	MAE	MAPE	RE
B0005	LSTM[10]	0.0786	0.0523	3.750%	0.9846
	SVR[11]	0.0678	0.0426	3.040%	0.9846
	KLMS[16]	0.0437	0.0363	2.481%	0.8154
	KRLS[17]	0.0601	0.0334	2.390%	0.8308
	EX-KRLS[19]	0.0444	0.0279	1.956%	0.8308
	SW-KRLS[18]	0.0359	0.0294	2.065%	0.8154
	QKRLS[20]	0.0395	0.0278	1.955%	0.8154
	FB-KRLS[21]	0.0265	0.0224	1.559%	0.9538
	SCKF-KRLS	0.0271	0.0196	1.390%	0.9692
	CKF-FB-KRLS	0.0182	0.0127	0.883%	0.9538
	SCKF-FB-KRLS	0.0161	0.0100	0.691%	0.9692
	DBN-SCKF-FB-KRLS	0.0068	0.0049	0.353%	1.0000

Table 7. Prediction results of different models for B0006 (60 Cycle).

No.	Method	RMSE	MAE	MAPE	RE
B0006	LSTM[10]	0.0759	0.0616	4.580%	0.8163
	SVR[11]	0.0657	0.0456	3.450%	0.9388
	KLMS[16]	0.0600	0.0403	3.021%	0.9388
	KRLS[17]	0.0473	0.0350	2.600%	0.9796
	EX-KRLS[19]	0.0462	0.0323	2.333%	0.9388
	SW-KRLS[18]	0.0415	0.0298	2.201%	0.8571
	QKRLS[20]	0.0363	0.0328	2.463%	0.8367
	FB-KRLS[21]	0.0330	0.0268	1.927%	0.9388
	SCKF-KRLS	0.0293	0.0249	1.872%	0.9388
	CKF-FB-KRLS	0.0213	0.0176	1.330%	0.9592
	SCKF-FB-KRLS	0.0143	0.0116	0.873%	0.9796
	DBN-SCKF-FB-KRLS	0.0097	0.0076	0.588%	0.9592

Table 8. Prediction results of different models for B0007 (60 Cycle).

No.	Method	RMSE	MAE	MAPE	RE
B0007	LSTM[10]	0.0778	0.0541	3.590%	0.9352
	SVR[11]	0.0658	0.0402	2.670%	--
	KLMS[16]	0.0773	0.0492	3.194%	--
	KRLS[17]	0.0574	0.0399	2.620%	0.9630
	EX-KRLS[19]	0.0402	0.0298	1.947%	--
	SW-KRLS[18]	0.0571	0.0303	2.020%	0.8704
	QKRLS[20]	0.0336	0.0311	2.059%	--
	FB-KRLS[21]	0.0331	0.0285	1.851%	--
	SCKF-KRLS	0.0307	0.0280	1.849%	--
	CKF-FB-KRLS	0.0265	0.0238	1.569%	--
	SCKF-FB-KRLS	0.0183	0.0145	0.954%	--
	DBN-SCKF-FB-KRLS	0.0029	0.0027	0.181%	--

Table 9. Prediction results of different models for B0018 (60 Cycle).

No.	Method	RMSE	MAE	MAPE	RE
B0018	LSTM[10]	0.0644	0.0453	3.200%	0.8108
	SVR[11]	0.0528	0.0413	2.910%	0.7838
	KLMS[16]	0.0544	0.0373	2.606%	0.9189
	KRLS[17]	0.0510	0.0365	2.570%	0.9189
	EX-KRLS[19]	0.0278	0.0200	1.400%	0.9459
	SW-KRLS[18]	0.0344	0.0209	1.474%	0.9189
	QKRLS[20]	0.0224	0.0190	1.346%	0.8649
	FB-KRLS[21]	0.0229	0.0185	1.289%	0.9459
	SCKF-KRLS	0.0191	0.0164	1.156%	0.9459
	CKF-FB-KRLS	0.0138	0.0102	0.709%	0.9459
	SCKF-FB-KRLS	0.0121	0.0083	0.584%	0.9730
	DBN-SCKF-FB-KRLS	0.0052	0.0046	0.327%	0.9730

Statistical analysis of prediction results when the prediction starting points are set at 80 cycles and 60 cycles, respectively, is shown in Table 2 - 9. The prediction outcomes for RUL from models trained using diverse algorithms across various lithium battery datasets are elaborated within. Particularly for battery B0007, if the predicted value falls short of the 70% failure threshold, the RE is denoted as "--".

Table 2 – 5 provide the prediction results for four batteries at a prediction starting point of 80 cycles, along with detailed performance evaluation metrics like RMSE, MAE, MAPE, and RE, which describe the fitting performance of lithium battery RUL predictions. For battery B0005, the DBN-SCKF-FB-KRLS method yielded an RMSE of 0.004, an MAE of 0.0037, and a MAPE of 0.26%. Additionally, predictive performance evaluations were conducted for batteries B0006, B0007, and B0018. The findings indicate that the DBN-SCKF-FB-KRLS approach consistently delivers lower error rates and superior accuracy relative to other methods. This underscores the effectiveness of the proposed method in this study for predicting the RUL of lithium batteries.

Table 6 - 9present the prediction results for four batteries at a prediction starting point of 60 cycles, along with detailed evaluation metrics like RMSE, MAE, MAPE, and RE. These metrics evaluate the performance of lithium battery RUL predictions. Notably, shifting the prediction starting point from 80 to 60 cycles results in a decrease in the accuracy of the lithium battery RUL forecasts. For battery B0005, the DBN-SCKF-FB-KRLS method resulted in an RMSE of 0.0068, an MAE of 0.0049, and a MAPE of 0.353%. Batteries B0006, B0007, and B0018 also showed similar results. This reinforces the advantages of the method outlined in this study for accurately predicting the RUL of lithium batteries.

4. Conclusions

Forecasting the RUL of lithium-ion batteries enables effective monitoring of their safety status, thereby ensuring the powered devices operate safely and stably. This paper addresses issues such as difficulty in data feature extraction, low prediction accuracy, and poor convergence effects encountered in the process of predicting the RUL of lithium batteries. The study introduces a RUL prediction model for lithium-ion batteries

utilizing the DBN-SCKF-FB-KRLS methodology. The model leverages the strengths of DBN in deep feature extraction to extract features from the original dataset, thereby reducing the complexity of the data. SCKF exhibits significant advantages in handling nonlinear system issues, while FB-KRLS has certain limitations in dealing with nonlinear systems. Therefore, combining the two creates the SCKF-FB-KRLS method, which synergizes their strengths to improve performance in complex scenarios. This method has significant advantages in handling nonlinear issues. The SCKF and FB-KRLS algorithms synergize effectively, and as the number of iterations increases, this approach significantly improves the accuracy of predicting the remaining life of lithium-ion batteries. On the other hand, the kernel adaptive filtering algorithm itself uses recursive iteration for parameter updating, which gives the advantage of real time processing of the dataset, and thus the problem of computation time due to the minimization of the error is mitigated to a certain extent.

The comparison of prediction results across different batteries and various starting points, alongside a review of existing literature, shows that the kernel adaptive filtering model DBN-SCKF-FB-KRLS presented in this paper enhances the prediction performance of lithium-ion batteries' remaining life, while maintaining stability. This ensures the safe and stable functioning of lithium-ion batteries.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (51467008), Gansu Provincial Department of Education Industry Support Program (2021CYZC-32), Gansu Provincial Science and Technology Program (23JRRA892, 24JRRA243).

References

1. Alsuwian T, Ansari S, Zainuri MA, Ayob A, Hussain A, Lipu MH, Alhawari AR, Almawgani AH, Almasabi S, Hindi AT. A Review of Expert Hybrid and Co-Estimation Techniques for SOH and RUL Estimation in Battery Management System with Electric Vehicle Application. Expert Systems with Applications. 2024 Jan 4:123123, https://doi.org/10.1016/j.eswa.2023.123123.
2. Ge MF, Liu Y, Jiang X, Liu J. A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries. Measurement. 2021 Apr 1;174:109057, https://doi.org/10.1016/j.measurement.2021.109057.
3. Liu F, Yu D, Shao C, Liu X, Su W. A review of multi-state joint estimation for lithium-ion battery: Research status and suggestions. Journal of Energy Storage. 2023 Dec 15;73:109071, https://doi.org/10.1016/j.est.2023.109071.
4. Ning J, Xiao B, Zhong W, Xiao B. A rapid detection method for the battery state of health. Measurement. 2022 Feb 15;189:110502, https://doi.org/10.1016/j.measurement.2021.110502.
5. Rahimi-Eichi H, Ojha U, Baronti F, Chow MY. Battery management system: An overview of its application in the smart grid and electric vehicles. IEEE industrial electronics magazine. 2013 Jun 14;7(2):4-16. https://doi.org/10.1109/MIE.2013.2250351
6. Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, Chen MH, Aykol M, Herring PK, Fraggedakis D, Bazant MZ. Data-driven prediction of battery cycle life before capacity degradation. Nature Energy. 2019 May;4(5):383-91. https://doi.org/10.1038/s41560-019-0356-8
7. Safavi V, Mohammadi Vaniar A, Bazmohammadi N, Vasquez JC, Guerrero JM. Battery Remaining Useful Life Prediction Using Machine Learning Models: A Comparative Study. Information. 2024 Feb 22;15(3):124. https://doi.org/10.3390/info15030124
8. Wang, Z., Shangguan, W., Peng, C., and Cai, B.. Similarity Based Remaining Useful Life Prediction for Lithium-ion Battery under Small Sample Situation Based on Data Augmentation. Eksploatacja i Niezawodność – Maintenance and Reliability, 2024 26(1). https://doi.org/10.17531/ein/175585.
9. Fang P, Sui X, Zhang A, Wang D, Yin L. Fusion model based RUL prediction method of lithium-ion battery under working conditions. Eksploatacja i Niezawodność – Maintenance and Reliability. 2024;26(3). https://doi.org/10.17531/ein/186537
10. Wang Z, Liu N, Chen C, Guo Y. Adaptive self-attention LSTM for RUL prediction of lithium-ion batteries. Information Sciences. 2023 Jul 1;635:398-413. https://doi.org/10.1016/j.ins.2023.01.100
11. Li X, Ma Y, Zhu J. An online dual filters RUL prediction method of lithium-ion battery based on unscented particle filter and least squares support vector machine. Measurement. 2021 Nov 1;184:109935, https://doi.org/10.1016/j.measurement.2021.109935.

12. Al-Greer M, Bashir I. Physics-based model informed smooth particle filter for remaining useful life prediction of lithium-ion battery. Measurement. 2023 Jun 15;214:112838. https://doi.org/10.1016/j.measurement.2023.112838
13. Li C, Zhang H, Ding P, Yang S, Bai Y. Deep feature extraction in lifetime prognostics of lithium-ion batteries: Advances, challenges and perspectives. Renewable and Sustainable Energy Reviews. 2023 Sep 1;184:113576, https://doi.org/10.1016/j.rser.2023.113576.
14. Liu D, Li L, Song Y, Wu L, Peng Y. Hybrid state of charge estimation for lithium-ion battery under dynamic operating conditions. International Journal of Electrical Power & Energy Systems. 2019 Sep 1;110:48-61, https://doi.org/10.1016/j.ijepes.2019.02.046.
15. Guo X, Ou S, Jiang M, Gao Y, Xu J, Cai Z. A New Sparse Kernel RLS Algorithm for Identification of Nonlinear Systems. IEEE Access. 2021 Dec 6;9:163165-77, https://doi.org/10.1109/ACCESS.2021.3133012.
16. Liu W, Pokharel PP, Principe JC. The kernel least-mean-square algorithm. IEEE Transactions on signal processing. 2008 Jan 16;56(2):543-54, https://doi.org/10.1109/TSP.2007.907881.
17. Engel Y, Mannor S, Meir R. The kernel recursive least-squares algorithm. IEEE Transactions on signal processing. 2004 Jul 19;52(8):2275-85, https://doi.org/10.1109/TSP.2004.830985.
18. Van Vaerenbergh S, Via J, Santamaría I. A sliding-window kernel RLS algorithm and its application to nonlinear channel identification. In2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings 2006 May 14 (Vol. 5, pp. 789-792). IEEE, https://doi.org/10.1109/ICASSP.2006.1661394.
19. Liu W, Park I, Wang Y, Principe JC. Extended kernel recursive least squares algorithm. IEEE Transactions on Signal Processing. 2009 May 2;57(10):3801-14, https://doi.org/10.1109/TSP.2009.2022007.
20. Chen B, Zhao S, Zhu P, Principe JC. Quantized kernel recursive least squares algorithm. IEEE transactions on neural networks and learning systems. 2013 May 13;24(9):1484-91, https://doi.org/10.1109/TNNLS.2013.2258936.
21. Van Vaerenbergh S, Santamaría I, Liu W, Príncipe JC. Fixed-budget kernel recursive least-squares. In2010 IEEE International Conference on Acoustics, Speech and Signal Processing 2010 Mar 14 (pp. 1882-1885). IEEE, https://doi.org/10.1109/ICASSP.2010.5495350.
22. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural computation. 2006 Jul 1;18(7):1527-54, https://doi.org/10.1162/neco.2006.18.7.1527
23. Cheng Y, Wan S, Choo KK. Deep belief network for meteorological time series prediction in the Internet of Things. IEEE internet of things journal. 2018 Oct 28;6(3):4369-76, https://doi.org/10.1109/JIOT.2018.2878477
24. Hu S, Xiang Y, Huo D, Jawad S, Liu J. An improved deep belief network based hybrid forecasting method for wind power. Energy. 2021 Jun 1;224:120185, https://doi.org/10.1016/j.energy.2021.120185.
25. Cao M, Zhang T, Wang J, Liu Y. A deep belief network approach to remaining capacity estimation for lithium-ion batteries based on charging process features. Journal of Energy Storage. 2022 Apr 1;48:103825, https://doi.org/10.1016/j.est.2021.103825.
26. Sun X, Wang G, Xu L, Yuan H, Yousefi N. Optimal estimation of the PEM fuel cells applying deep belief network optimized by improved archimedes optimization algorithm. Energy. 2021 Dec 15;237:121532, https://doi.org/10.1016/j.energy.2021.121532.
27. Arasaratnam I, Haykin S. Cubature kalman filters. IEEE Transactions on automatic control. 2009 May 27;54(6):1254-69, https://doi.org/10.1109/TAC.2009.2019800.
28. Han M, Xu M, Liu X, Wang X. Online multivariate time series prediction using SCKF-γESN model. Neurocomputing. 2015 Jan 5;147:315-23, https://doi.org/10.1016/j.neucom.2014.06.057.