Cuckoo Search Based Relevance Vector Machine with Hybrid Kernel for Battery Remaining Useful Life Prediction

Yiting Chen; Chun Zhang; Ning Zhang; Xue Guo; Huan Wang; Yaohua Chen
Engineering Research Center of Network Management Technology for High Speed Railway of Ministry of Education
Beijing Jiaotong University
Beijing, China
17120350@bjtu.edu.cn

***Abstract*—The study of lithium-ion battery management has always been a hot issue in energy field. Predicting the remaining useful life (RUL) helps to know the time of maintenance and replacement in advance so that users can be avoided from unexpected battery fault. In this paper, a RUL prediction method on the basis of a sparse probability model in the Bayesian framework which is Relevance Vector Machine (RVM) is proposed. Hybrid kernel function is introduced to ensure the prediction accuracy. A newly heuristic algorithm Cuckoo Search (CS) is adopted to intelligently determine all the kernel parameters along with the weight of each kernel function. Contrast experiments show that our proposed approach based on hybrid kernel RVM with CS (CS-HKRVM) outperforms the basic machine learning algorithms including single kernel RVM, Back Propagation Neural Network and Support Vector Machine. The hybrid kernel RVM with particle swarm optimization (PSO) is also conducted in order to demonstrate that CS is more effective and accurate than PSO in searching for the optimal solution of prediction model construction.**

Keywords-lithium-ion battery; relevance vector machine; hybrid kernel; remaining useful life prediction; cuckoo search

I. INTRODUCTION

Nowadays, there has been widespread use of lithium-ion batteries as power sources in a broad variety of fields such as aerospace and rail transit, owing to their good features of high energy density, environmental protection, high security and long lifetime [1]. In the process of practical application, the performance of lithium-ion batteries gradually degrades during charge-discharge cycles. And once the available capacity fades to a given threshold, failure is judged to occur, which could result in the decreasing efficiency of the whole system. Predicting the battery remaining useful life (RUL) [2] helps to learn the degradation process so that users are capable of making advanced maintenance decision instead of suffering loss from unexpected fault. The battery RUL in our paper is defined as the remaining cycle life when the available capacity reaches 70% of the rated capacity.

Recently the mainstream approaches for RUL prediction are categorized into the following types: model-based methods, data-driven methods and fusion methods [3-5]. Methods based on model analyze the performance degradation and failure mechanism of target objects to establish the mathematical and physical model. The complexity of lithium-ion battery system, noise and environmental uncertainty make the precise and universal model hard to build. In comparison, data-driven methods including support vector machine (SVM), Gaussian process (GP), relevance vector machine (RVM), autoregressive integrated moving average (ARIMA), artificial neural network (ANN) and so on, only depend on the historical records and extract the key information to construct the degradation model. However, the prediction accuracy cannot be guaranteed. For the sake of boosting the prediction precision of data-driven methods, the fusion methods which consist of multiple data-driven algorithms have been proposed. Chen et al. [6] came up with a hybrid method which combines Variational mode decomposition (VMD) denoising technique, ARIMA, and GM (1,1) models for RUL prediction. Zhang et al. [7] chose a deep-learning method to improve the long-term learning performance of RUL prediction. A Gauss-Hermite particle filter (GHPF) was brought up to realize the real-time updating of model parameters and prediction of battery RUL by Ma et al. [8]. All the above methods have achieved good regression performance, but it is inconvenient to decide their best parameters to ensure the highest accuracy. Zhang et al. [9] exploited particle swarm optimization (PSO) to calculate the most suitable weights of multiple kernel functions of RVM to make accurate predictions about the future capacity of the battery.

In this paper, a novel optimization approach for hybrid kernel RVM is proposed to provide high-precision model. RVM is a sparse probability model under the Bayesian Framework and it owns the ability of uncertainty expression. The kernel function of RVM is avoided from meeting the Mercer condition so the options are quite open. However, using single kernel function can lead to weak generalization capability. To solve this problem, combined kernel is put to use

This work was sponsored by the China Railway Corporation Program "Overall Technology Research of PHM for EMU"(No.K2018J018).

978-1-7281-0860-5/19/$31.00 ©2019 IEEE

and the weight, along with all the parameters of each kernel function, is designed to be intelligently determined by a heuristic algorithm Cuckoo Search (CS) [10]. Utilizing Lévy Flight, CS algorithm aims at effectively solving the optimization problem, thus seeking for the best training model to get the most accurate prediction results.

The arrangement of the rest of this paper follows: Section 2 introduces the theories of RVM, hybrid kernel function and CS algorithm. The experimental data are illustrated in Section 3 along with our proposed RUL prediction approach based on hybrid kernel RVM with CS (CS-HKRVM). Section 4 shows the effectiveness and accuracy of our method and makes comparisons with the performance of several existing algorithms. Ultimately, we come to a conclusion in Section 5.

II. RELATED THEORY

A. Relevance Vector Machine Regression

RVM is a modeling technology in the Bayesian framework utilizing existing data and was first proposed by Tipping [11] in the basis of SVM. Assuming a training sample set $\{x_n, z_n\}_{n=1}^{N}$, where N marks the number of the training data, $x_n \in R^d$, $z_n \in R$. The target z_n can be formulated as:

$$z_n = y(x_n;\boldsymbol{\omega}) + \epsilon \tag{1}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_N)^T$ represents the corresponding weights, and ϵ denotes independent noise subject to $N(0, \sigma^2)$. The likelihood function of the training sample set can be calculated in the condition of assuming that z_n is independent by the following formula:

$$p(\mathbf{z} \mid \boldsymbol{\omega}, \sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left(-\frac{\| \mathbf{z} - \boldsymbol{\Phi}\boldsymbol{\omega} \|^2}{2\sigma^2}\right) \tag{2}$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_N)^T$, $\boldsymbol{\Phi} = [\phi(x_1), \phi(x_2), \ldots, \phi(x_N)]^T$ is the kernel function matrix with $\phi(x_n) = [1, K(x_n, x_1), K(x_n, x_2), \ldots, (x_n, x_N)]^T$, and $K(x_n, x)$ is kernel.

To avoid over-fitting that is brought by maximum likelihood estimations of $\boldsymbol{\omega}$ and σ^2 in (2), RVM applies a Gaussian prior distribution to constraint the parameter vector $\boldsymbol{\omega}$ and it is expressed as:

$$p(\boldsymbol{\omega} \mid \boldsymbol{\alpha}) = \prod_{i=0}^{N} \frac{\alpha_i}{\sqrt{2\pi}} \exp\left(-\frac{\alpha_i \omega_i^2}{2}\right) \tag{3}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T$ is a vector of N+1 hyper-parameters and every hyper-parameter is separately bound up with one weight to get a sparse model. $\boldsymbol{\alpha}$ obeys Gamma distribution with the noise variance σ^2. According to the Bayes rule, the posterior probability of the whole number of unknowns from sample data can be obtained by:

$$p(\boldsymbol{\omega}, \boldsymbol{\alpha}, \sigma^2 \mid \mathbf{z}) = \frac{p(\mathbf{z} \mid \boldsymbol{\omega}, \boldsymbol{\alpha}, \sigma^2)\, p(\boldsymbol{\omega}, \boldsymbol{\alpha}, \sigma^2)}{p(\mathbf{z})} \tag{4}$$

If a new vector x_* is input, the corresponding prediction z_* under a sparse Bayesian learning framework is computed by:

$$p(z_* \mid \mathbf{z}) = \int p\left(z_* \mid \boldsymbol{\omega}, \boldsymbol{\alpha}, \sigma^2\right) p\left(\boldsymbol{\omega}, \boldsymbol{\alpha}, \sigma^2\right) d\boldsymbol{\omega} d\boldsymbol{\alpha} d\sigma^2 \tag{5}$$

However, (5) is hardly soluble, (4) can be replaced by:

$$p(\boldsymbol{\omega}, \boldsymbol{\alpha}, \sigma^2 \mid \mathbf{z}) = p(\boldsymbol{\omega} \mid \mathbf{z}, \boldsymbol{\alpha}, \sigma^2)\, p(\boldsymbol{\alpha}, \sigma^2 \mid \mathbf{z}) \tag{6}$$

And the posterior distribution of the weights can be generated as:

$$\begin{aligned} &p(\boldsymbol{\omega} \mid \mathbf{z}, \boldsymbol{\alpha}, \sigma^2) \\ &= (2\pi)^{-\frac{N+1}{2}} \mid \boldsymbol{\Sigma} \mid^{-\frac{1}{2}} \exp\left\{-\frac{(\boldsymbol{\omega} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\omega} - \boldsymbol{\mu})}{2}\right\} \end{aligned} \tag{7}$$

where $\boldsymbol{\Sigma} = (\mathbf{A} + \sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}$ is the posterior covariance and $\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{z}$ stands for the mean of weight, $\mathbf{A} = diag(\alpha_0, \alpha_1, \ldots, \alpha_N)$. Then for the hyper-parameters, their marginal likelihood is calculated by:

$$p(\mathbf{z} \mid \boldsymbol{\alpha}, \sigma^2) = \int p(\mathbf{z} \mid \boldsymbol{\omega}, \sigma^2)\, p(\boldsymbol{\omega} \mid \boldsymbol{\alpha}) d\boldsymbol{\omega} \tag{8}$$

The weights' maximum posterior (MP) estimation relies on the value of $\boldsymbol{\alpha}$ and σ^2. To maximize the marginal likelihood, we set the required derivatives to zero and obtain $\boldsymbol{\alpha}_{MP}$ and σ^2_{MP} by iterative formulas:

$$\alpha_i^{new} = \frac{1 - \alpha_i N_{ii}}{\mu_i^2} \tag{9}$$

$$(\sigma^2)^{new} = \frac{\| \mathbf{z} - \boldsymbol{\Phi}\boldsymbol{\mu} \|^2}{N - \Sigma_i (1 - \alpha_i N_{ii})} \tag{10}$$

where N_{ii} means the *i*-th diagonal element in matrix $\boldsymbol{\Sigma}$.

During the process of iteration, there will be lots of hyper-parameters approaching infinity, and the corresponding weights tend to be zero. In RVM, those data samples whose weights are not equal to zero are called Relevance Vectors (RVs). Once a new vector x_* is input, the prediction z_* probability distribution can be expressed as follows:

$$p(z_* \mid \mathbf{z}, \boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) = \int p(z_* \mid z, \boldsymbol{\omega}, \sigma^2_{MP})\, p(\boldsymbol{\omega} \mid \mathbf{z}, \boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) d\boldsymbol{\omega} \tag{11}$$

$$p(\boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) = \arg\max_{\alpha, \sigma^2} p(\boldsymbol{\alpha}, \sigma^2 \mid \mathbf{z}) \tag{12}$$

Since both integrand terms are Gaussian, we get:

$$p(z_* \mid \mathbf{z}) \sim N(z_* \mid y_*, \sigma_*^2) \tag{13}$$

where $y_* = \mu^T \varphi(x_*)$, and $\sigma_*^2 = \sigma^2_{MP} + \phi(x_*)^T \boldsymbol{\Sigma} \phi(x_*)$.

B. Hybrid Kernel Function

In order to make accurate prediction, the proper choice of RVM kernel function is necessary. The introduction of kernel avoids the problem of arithmetical operation complexity in high-dimensional space by inner product operation. The related research work of RVM [12][13] has shown that single kernel

function has been already widely applied. The traditional kernel functions are divided into two major categories: local kernel and global kernel. Local kernel possesses the feature of powerful learning capability and weak generalization capacity. One typical local kernel is Gaussian kernel which owns strong nonlinear processing ability. And it can be described as:

$$K_1(x_n, x) = \exp(-\frac{\| x_n - x \|^2}{2\gamma^2}) \tag{14}$$

where γ is the width factor.

By contrast, global kernel takes dot product function of all the samples in feature space into consideration, thus gaining excellent generalization ability, which forms a good mutual complementary relationship with local kernel. Polynomial kernel has been applied in lots of machining learning research work [14][15] and the function can be described as:

$$K_2(x_n, x) = [(x_n^T \cdot x) + 1]^d \tag{15}$$

where d refers to the degree.

Combining the advantages of these two kinds of kernel, this more novel hybrid kernel way tends to give the prediction of battery capacity in the future with high-precision and it is defined as:

$$K(x_n, x) = bK_1(x_n, x) + (1-b)K_2(x_n, x) \tag{16}$$

where b represents the weight of the Gaussian kernel while $(1-b)$ stands for the weight of polynomial kernel function. Now the parameters need to be set are the width factor γ, the degree d and the weight b. CS algorithm is employed to find the optimal parameters combination.

C. Cuckoo Search Algorithm

Yang et al. [10] proposed a meta-heuristic algorithm inspired by cuckoo's nest parasitism and Lévy Flight mode that is known as Cuckoo Search Algorithm. The cuckoo bird family with no incubation behavior tend to lay eggs in their chosen host nests so that the host birds hatch the mistaken eggs. A cuckoo may remove the host eggs to keep the number of eggs invariant, increasing the hatching probability of their own children. After the baby cuckoo is born, it will imitate the cry of the host bird, even ousting other host babies. However, once the host birds find out the eggs do not belong to them, they will immediately abandon the original nest or throw out the wrong eggs. The CS algorithm is to model the behavior of a cuckoo searching for the optimal nest for its egg by random walk, thus obtaining the optimum solution of the target problem [16].

Three idealized rules are introduced to simply depict the standard CS:

(1) Each time a cuckoo lays only one egg and it randomly chooses a nest to incubate the egg.

(2) The finest nest will be reserved for the next generation.

(3) The amount of available host nests is fixed and the thing that the cuckoo egg is discovered and abandoned is given a probability p_a, and $p_a \in [0,1]$.

The CS parameters which consist of the bird nest amount N, the step size α, searching space dimension dim, p_a and the maximum amount of iteration times are set in the first step. Under these circumstances, the updating formula of nest location and search path is defined as:

$$nest_i^{t+1} = nest_i^t + \alpha \oplus L(\lambda) \tag{17}$$

where $nest_i^t$ represents the location of the i-th nest at iteration t, α is the step size, $\oplus$ means entrywise multiplications and $L(\lambda)$ is the Lévy random search path.

Once the location of bird nests is updated, the generated random number $r \in [0,1]$ is compared with p_a. When $r < p_a$, several bad nests will be abandoned and the same number of new nests will be generated. The optimum position of nest can be reserved.

III. RUL PREDICTION BASED ON CS-HKRVM

A. Experimental Data

We adopt the open lithium-ion battery data set from the NASA Ames Prognostics Center of Excellence (PCoE) [17] to validate the effectiveness of our proposed method. During the data collection procedure, tests are run over the batteries at room temperature in three different operational processes: charge, discharge and impedance. During the charging period, the batteries were loaded with a constant current of 1.5A and the charging process stopped when the voltage increased to 4.2V, then the voltage was maintained until the charging current became 20mA. Discharging test was conducted at a constant current of 2A until the cutoff voltage fell to the predefined value.

The No.5 and No.18 battery data have been chosen as our research experimental objects, and their cutoff voltages are respectively 2.7V and 2.5V. The measured capacity data along with the cycle times are illustrated in Fig. 1.

One cycle means one complete charge and discharge operation. The experiments were finished once the capacity dropped from 2Ah to 1.4Ah. And in this experiment, the sizes of No.5 and No.18 capacity data are 168 cycles and 132 cycles.

B. Hybrid Model for Battery RUL

The steps of battery RUL prediction based on CS optimized hybrid kernel RVM are as follows:

Step 1: Analyze the lithium-ion battery data set from NASA and select discharge voltage degradation data along with capacity data. Calculate the average terminal voltage and discharging time under the same voltage in each cycle and take them as the original input data while the output is capacity. Determine the training samples and testing samples. Preprocess all the sample data for normalization to improve the calculation speed and prediction accuracy.

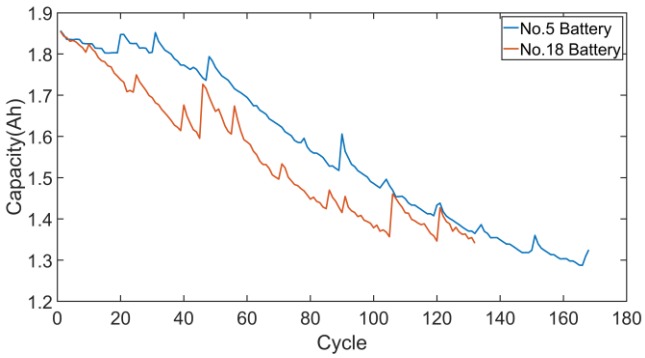


Figure 1. Capacity data for No.5 and No.18 battery.

Step 2: Implement parameter optimization with CS algorithm and train the model. Set Mean Squared Error (MSE) of the training prediction results as the fitness function of CS. Aiming at minimizing MSE, we find out the most suitable parameter combination of the width factor γ, the degree d and the weight b through continued iterations. Using the optimal parameters to predict the test data could achieve the highest accuracy.

Step 3: Perform the model prediction after the training process is finished. The testing data are input to the optimized model to verify the reliability and effectiveness of our method. And the prediction data will be re-normalized.

The specific workflow of the parameter optimization with CS algorithm is illustrated as:

Step 1: Set the range of each targeted RVM parameter and initialize all the CS algorithm parameters including the bird nest amount N, the step size α, searching space dimension dim, p_a and the maximum amount of iteration times $t_{\max}$.

Step 2: Initialize the nest positions randomly by formula $q_i^0 = \left(x_1^0, x_2^0, \ldots\right.$. Calculate the fitness value of each nest, taking the position which corresponds with the minimum fitness value as the current best solution. The fitness function is defined as:

$$MSE = \frac{1}{ntr}\sum_{i=1}^{ntr}(Y_i - Y_i)^2 \tag{18}$$

where ntr means the amount of the training data, Y_i is the real capacity value and Y_i means the model prediction result.

Step 3: Keep the former best solution and update the position of the remaining nests according to Formula (17) and calculate the fitness value of the new position. If this new fitness value is smaller than that of the previous generation, replace the old solution with the new one.

Step 4: Given a random number r, compare r with p_a and if $r > p_a$, randomly change the position of nests with relatively higher discovering probability to get a new set of nest positions.

Step 5: Return to **Step 3** until the maximum iteration limit is reached and output the best set of parameters (γ, d, b) to establish the optimal RVM model.

IV. RESULT AND DISCUSSION

A. *Experimental Environment*

Our experiment was implemented on Matlab_R2018b of a desktop computer equipped with a 3.20GHz Intel Core i5 processor and 12.0 GB RAM. The initial parameters of the CS algorithm were set in advance. N=20, p_a=0.25, $t_{\max}$=50 and dim=3. The ranges of γ and d are both from 0.1 to 20, while b is from 0.01 to 0.99.

B. *Evaluation Criteria*

We performed the RUL prediction at different starting point for No.5 and No.18 battery. Two functions were used to evaluate the effectiveness and accuracy of our method.

1. Absolute Error (AE)

$$AE = | R - \hat{R} | \tag{19}$$

2. Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{nte}\sum_{i=1}^{nte}(Ytest_i - Ytest_i)^2} \tag{20}$$

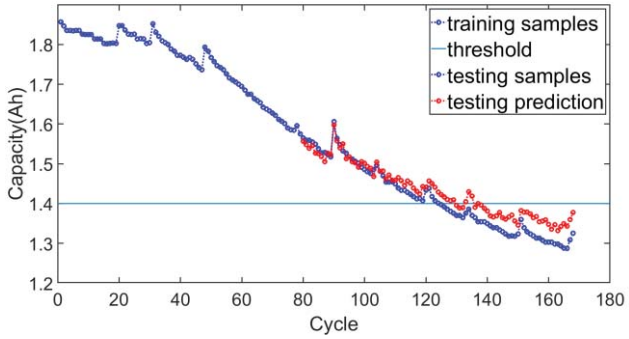
where nte means the amount of testing samples, $Ytest_i$ is the real capacity and $Ytest_i$ means the testing prediction result.

The methods of contrast experiments consist of Gaussian RVM, SVM, back propagation neural network (BPNN), and the hybrid kernel RVM optimized by PSO (PSO-HKRVM).

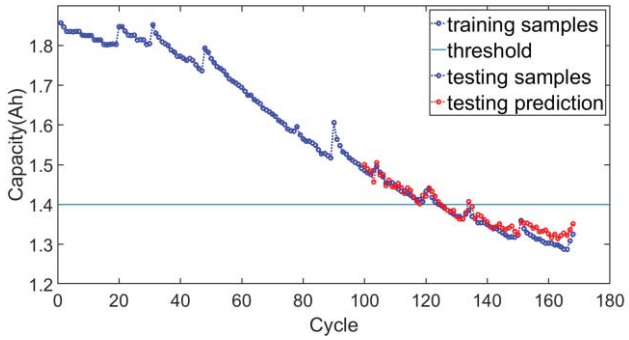
C. *Prediciton Result and Analysis*

The predicted RUL of No.5 battery at various starting points are shown in Fig. 2. Fig. 2(a) illustrates the experimental result of starting prediction at 80 cycles, while Fig. 2(b) depicts the experimental result of starting prediction at 100 cycles. The number of battery life cycles are plotted on the horizontal X-axis along with battery capacities on the vertical Y-axis. Sample data are drawn with blue dots and the testing prediction data are represented in red.

The predicted RUL of No.18 battery at various starting points are shown in Fig. 3. Fig. 3(a) shows the experimental result of starting prediction at 60 cycles, while Fig. 3(b) indicates the experimental result of starting prediction at 80 cycles.

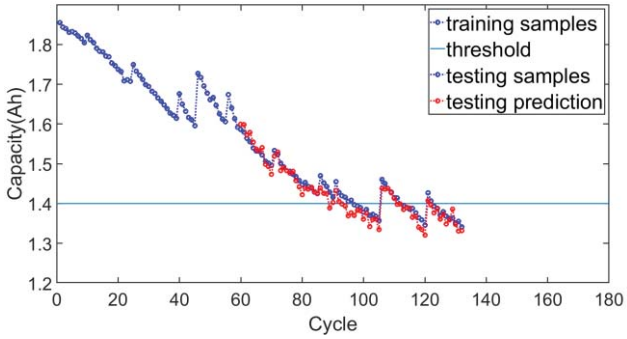


(a) Starting prediction at 80 cycles

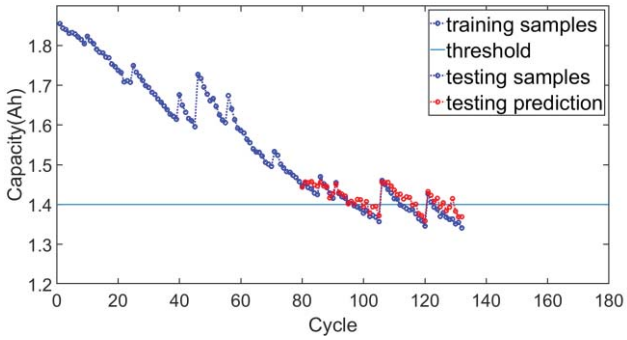


(b) Starting prediction at 100 cycles

Figure 2. No.5 battery RUL prediction results.



(a) Starting prediction at 60 cycles



(b) Starting prediction at 80 cycles

Figure 3. No.18 battery RUL prediction results.

Table I and Table II give the RUL prediction results of different contrast experiments with the mentioned evaluation criteria AE and RMSE for battery No.5 and No.18 respectively. The basic algorithms such as RVM, SVM and BPNN without optimization show relatively large errors which range from 9 to 22 for AE and from 0.0286 to 0.1982 for RMSE. The hybrid kernel RVM fused with PSO algorithm performs better than the above basic algorithms owing to its characteristic of approximating the optimal solution fast. But our proposed CS-HKRVM is proved to outperform all the contrast algorithms with the smallest AE and RMSE for its advantage of efficient global and local searching ability. The determined sets of parameters are S1=(0.2,1.0,0.24), S2=(8.0,8.0,0.89), S3=(1.0,1.9,0.10), S4=(1.0,12.1,0.90), where S1 and S2 are for the battery No.5 starting prediction at the 80-th cycle and the 100-th cycle respectively, S3 and S4 are for the battery No.18 starting prediction at the 60-th cycle and the 80-th cycle respectively.

According to the above analysis, it can be concluded that our proposed RVM model with hybrid kernel optimized by CS algorithm for lithium-ion battery RUL prediction possesses a combination with effectiveness, reliability and high accuracy.

V. CONCLUSIONS

In this paper, a HKRVM approach has been proposed to predict the lithium-ion battery RUL. Meanwhile, the parameters and weight of each kernel function are produced by CS algorithm to obtain the best set of model parameters, enhancing the prediction accuracy. We make use of the open battery data of No.5 battery and No.18 battery from NASA PCoE to verify the RUL prediction ability of our method.

TABLE I. EXPERIMENTAL RESULTS OF NO.5 BATTERY

Approach	Starting Point	Actual RUL	Predicted RUL	AE	RMSE
RVM	80	45	27	18	0.0714
	100	25	15	10	0.0518
SVM	80	45	59	14	0.0553
	100	25	36	11	0.0450
BPNN	80	45	66	21	0.1079
	100	25	37	12	0.0622
PSO-HKRVM	80	45	35	10	0.0496
	100	25	18	7	0.0340
CS-HKRVM	80	45	50	5	0.0274
	100	25	26	1	0.0169

TABLE II. EXPERIMENTAL RESULTS OF NO.18 BATTERY

Approach	Starting Point	Actual RUL	Predicted RUL	AE	RMSE
RVM	60	37	20	17	0.1055
	80	17	26	9	0.0655
SVM	60	37	17	20	0.1677
	80	17	6	11	0.0737
BPNN	60	37	59	22	0.1982
	80	17	30	13	0.0610
PSO-HKRVM	60	37	49	12	0.0766
	80	17	25	8	0.0286
CS-HKRVM	60	37	29	8	0.0141
	80	17	20	3	0.0189

What's more, the contrast experiments have proven that our proposed CS-HKRVM can achieve better prediction performance than RVM, SVM, BPNN and PSO-HKRVM in terms of AE and RMSE. The hybrid kernel has been proven to be superior to the single kernel in the experiment due to the combination of generalization and learning ability. The CS algorithm shows powerful advantages over the PSO algorithm in ensuring the global convergence so that we can get the optimal prediction model.

The future research work will be the RUL prediction for the battery in practical application with complex working conditions and the prediction method when we deal with mass data of other kinds of battery.

ACKNOWLEDGMENT

This work was supported by the China Railway Corporation Program "Overall Technology Research of PHM for EMU" (No. K2018J018).

REFERENCES

[1] M. A. Hannan, M. M. Hoque, A. Hussain, Y. Yusof and P. J. Ker, "State-of-the-art and energy management system of lithium-ion batteries in electric vehicle applications: issues and recommendations," IEEE Access, vol. 6, pp. 19362-19378, 2018.

[2] S. M. Rezvanizaniani, Z. Liu, Y. Chen and J. Lee, "Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle (EV) safety and mobility," Journal of Power Sources, vol. 256, pp. 110-124, 2014.

[3] L. Liao and F. Kottig, "Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction," IEEE Transactions on Reliability, vol. 63, pp.191-207, 2014.

[4] C. Su and H. J. Chen, "A review on prognostics approaches for remaining useful life of lithium-ion battery," IOP Conference Series: Earth and Environmental Science, vol. 93, pp. 012040, 2017.

[5] X. Xu, C. Yu, S. Tang, X. Sun, X. Si and L. Wu, "State-of-Health Estimation for Lithium-ion Batteries Based on Wiener Process with Modeling the Relaxation Effect," in IEEE Access, 2019.

[6] L. Chen, L. Xu and Y. Zhou, "Novel Approach for Lithium-Ion Battery On-Line Remaining Useful Life Prediction Based on Permutation Entropy," Energies, vol. 11, pp. 820, 2018.

[7] Y. Zhang, R. Xiong, H. He and M. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," IEEE Transactions on Vehicular Technology, vol. 67, pp.5695-5705, 2018.

[8] Y. Ma, Y. Chen, X. Zhou and H. Chen, "Remaining useful life prediction of lithium-ion battery based on Gauss-Hermite particle filter," in IEEE Transactions on Control Systems Technology, pp.1-8, 2018.

[9] C. Zhang, Y. He, L. Yuan and S. Xiang, "Capacity prognostics of Lithium-ion batteries using EMD denoising and multiple kernel RVM," in IEEE Access, vol. 5, pp. 12061-12070, 2017.

[10] X. S. Yang, S. Deb, "Cuckoo search via Levy flights", Proceedings of World Congress on Nature & Biologically Inspired Computing, pp. 210-214, 2009.

[11] M. E. Tipping, J. Mach, "Sparse bayesian learning and the relevance vector machine," Learn. Res., vol. 1, pp. 211-244, Sep. 2001.

[12] F. Liu, J. Z. Zhou, F. P. Qiu, J. J. Yang, L. Liu, "Nonlinear hydrological time series forecasting based on the relevance vector regression", Neural Information Processing, 13th International Conference, ICONIP 2006, Hong Kong, China, October 3-6, 2006, Proceedings, Part II. DBLP, 2006.

[13] Jin Yuan, Tao Yu, Kesheng Wang and Xuemei Liu, "Adaptive Spherical Gaussian Kernel for fast Relevance Vector Machine Regression," 2008 7th World Congress on Intelligent Control and Automation, Chongqing, 2008, pp. 2071-2076.

[14] Z. Zhang and K. K. Parhi, "Seizure prediction using polynomial SVM classification," 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, 2015, pp. 5748-5751.

[15] M. Hussain, S. K. Wajid, A. Elzaart and M. Berbar, "A Comparison of SVM Kernel Functions for Breast Cancer Detection," 2011 Eighth International Conference Computer Graphics, Imaging and Visualization, Singapore, 2011, pp. 145-150.

[16] P. Civicioglu, E. Besdok. "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," Artificial Intelligence Review, vol. 39, pp. 315-346, 2013.

[17] B. Saha, K. Goebel, Battery Data Set. NASA Ames Prognostics Data Repository, 2007, [online] Available: http://ti.arc.nasa.gov/project/prognostic-data-repository.