Reliability Engineering and System Safety 000 (2018) 1–13

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction

Zhixiong Li [a], Dazhong Wu [b], Chao Hu [a,c,*], Janis Terpenny [d]

[a] Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA
[b] Department of Mechanical and Aerospace Engineering, University of Central Florida, Orlando, FL 32816, USA
[c] Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA
[d] Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, PA 16802, USA

ARTICLE INFO

Keywords:
Degradation-dependent weights
Remaining useful life
Prognostics
Ensemble learning
Locally weighted regression

ABSTRACT

Remaining useful life (RUL) prediction is crucial for the implementation of predictive maintenance strategies. While significant research has been conducted in model-based and data-driven prognostics, there has been little research reported on the RUL prediction using an ensemble learning method that combines prediction results from multiple learning algorithms. The objective of this research is to introduce a new ensemble prognostics method that takes into account the effects of degradation on the accuracy of RUL prediction. Specifically, this method assigns an optimized, degradation-dependent weight to each learner (i.e., learning algorithm) such that the weighted sum of the prediction results from all the learners predicts the RULs of engineered systems with better accuracy. The ensemble prognostics method is demonstrated using two case studies. One case study is to predict the RULs of aircraft bearings; the other is to predict the RULs of aircraft engines. The numerical results have shown that the predictive model trained by the ensemble learning-based prognostic approach with degradation-dependent weights is capable of outperforming the original ensemble learning-based approach and its member algorithms.

1. Introduction

Prognostics refers to "an estimation of time to failure and risk for one or more existing and future failure modes" [1]. In order to effectively support decision-making processes for condition-based maintenance (CBM), it is critical to predict the remaining useful life (RUL) of an engineered system [2]. A large number of prognostic methods have been developed over the past two decades [3]. In general, prognostic methods fall into three categories: (1) model-based, (2) data-driven, and (3) hybrid methods [4]. Model-based prognostics refers to the methods that use models derived from first principles or probability theory. For example, particle filters (PFs) [5,6], Kalman filter [7], and hidden Markov model [8] are model-based methods that leverage degradation models derived based on first principles (i.e., underlying degradation mechanisms) or probability theory. The prediction accuracy of model-based methods depends on the prior knowledge of physical behavior [4]. However, the domain knowledge for complex systems are not always available or too expensive to acquire. To complement model-based methods, data-driven methods refer to the methods that use models learned exclusively from data. Typical data-driven methods use interpolation [9], extrapolation [10], and machine learning [11] for RUL prediction. In data-driven methods, training data are used to design and train a predictive model; testing data are used to validate the predictive model. Data-driven methods are typically more effective than model-based methods for complex engineered systems such as aircraft engines and wind turbines. Hybrid prognostics refers to the methods that facilitate the combined use of model-based and data-driven methods. Liao and Köttig [4] presented a comprehensive review on hybrid prognostics for RUL prediction. According to this review, hybrid prognostics shows great potential for achieving better prediction accuracy than model-based and data-driven prognostics. Ensemble learning-based prognostics (or ensemble prognostics) is among one of the most popular hybrid methods, and has been shown to be capable of improving prediction accuracy by combining multiple learning algorithms [12]. While much research has been conducted in the area of prognostics, there are still remaining challenges [13]. This paper focuses on addressing the following issue: How can the effects of time-dependent degradation be taken into account when predicting the RUL of an engineered system?

The main contributions of this research is that:

* Corresponding author at: Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA.
E-mail addresses: chaohu@iastate.edu, huchaostu@gmail.com (C. Hu).

https://doi.org/10.1016/j.ress.2017.12.016
Available online xxx
0951-8320/

Nomenclature

ANN	artificial neural network
CBM	condition-based maintenance
CV	cross validation
ES	similarity-based interpolation (SBI) with the least-square exponential fitting
EDI	ensemble prognostics with degradation-independent weights
EDD	ensemble prognostics with degradation-dependent weights
EOL	end of life
FPT	first predicting time
LWR	locally weighted linear regression
PF	particle filter
PSW	phase space warping
PHM	prognostics and health management
QB	Bayesian linear regression with the least-square quadratic fitting
RUL	remaining useful life
RVM	relevance vector machine
RS	SBI with RVM
RNN	recurrent neural network
SBI	similarity-based interpolation
SVM	support vector machine
SS	SBI with SVM
VHI	virtual health index

(1) An ensemble learning-based prognostic approach that accounts for the effects of time-dependent degradation is introduced.
(2) Two case studies were conducted to demonstrate the ensemble learning-based method with degradation-dependent weights.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the ensemble prognostics method. Section 4 discusses the experimental results. Section 5 provides conclusions and future work.

2. Related work

2.1. Ensemble learning

This section provides a brief overview of ensemble learning in RUL prediction. Ensemble-learning methods are learning algorithms that aggregate predictions produced by multiple learning algorithms in order to improve predictive performance [14]. Previous studies have shown that an ensemble learning algorithm typically outperforms any of the constituent learning algorithms alone [15]. Sun et al. [16] demonstrated the effectiveness of ensemble learning for estimating gas turbine engine degradation. In this study, a dynamic weight allocation method based on the Adaboost ensemble learning algorithm was introduced to train several multi-layer perceptron neural networks. Experimental results have shown that the ensemble learning-based fusion prognostic method can improve the prediction accuracy by 35% when compared with a single neural network model. Xing et al. [17] combined an empirical exponential and a polynomial regression model to estimate the RUL of lithium-ion batteries. The Monte Carlo simulation was used to calculate the ensemble weights. The proposed ensemble method was able to track the aging trend of the battery while either the single exponential model or the single polynomial model failed in the case study. Zhang et al. [18] introduced an ensemble learning method for predicting the RUL of rolling bearings. Two artificial neural networks were combined using a simple weight-vector. Baraldi et al. developed a PF-based prognostic approach that employs a bagged ensemble of artificial neural networks (ANNs) as an empirical measurement model in PF, and demonstrated this approach through a case study on the RUL prediction of an engineering structure [48]. The authors later proposed an ensemble method that aggregates four kernel regression models to estimate the health conditions of choke valves used in offshore oil platforms [19]. The analytic hierarchy process was employed to obtain the ensemble weights for the kernel regression models. In addition, another ensemble method that combines Gaussian process regression and similarity-based regression [20] was presented to improve prediction accuracy. Wu et al. [21] introduced a random forests-based prognostic approach to train a predictive model by aggregating a collection of regression trees. The effectiveness of this method was demonstrated using tool wear prediction. Experimental results have shown that the random forests-based prognostic method can generate predictions with very high accuracy. Sbarufatti et al. [49] trained a committee of ANNs with strain patterns simulated with a finite element model and used the trained ANNs to estimate the crack length in real time. The crack-length estimate was fed as an input (or measurement) into sequential Monte-Carlo simulation that produced the posterior distribution of the RUL of a structural component subjected to fatigue crack propagation. Cadini et al. [50] also employed a committee of trained ANNs to estimate the crack length of a structural component within a PF-based probabilistic framework, although their primary application was early crack diagnostics, not prediction of crack propagation and RUL. Ensemble learning methods have been applied to the estimation of aircraft engine performance. Peel [22] used the Kalman filter to combine prediction results produced by different ANNs. Experimental results have shown that this method outperforms each artificial network model using the prognostic data sets for the 2008 IEEE PHM Data Challenge Competition.

While ensemble learning has been applied in the field of PHM to enhance the accuracy of RUL prediction, only a few studies have introduced ensemble learning-based approaches that consider the effects of time-dependent degradation on the prognostic accuracy. For example, Peng and Dong [23] showed that by considering the deterioration of the pumps, the degradation-dependent prognostics approach produced more accurate RUL prediction than the degradation-independent one. Liu et al. [24] proposed a degradation-dependent autoregressive model to improve the RUL prediction accuracy for lithium-ion batteries. Lim et al. [25] proposed a degradation-dependent Kalman filter ensemble prognostics approach, which takes different degradation stages into account. Liu et al. [26] used dynamic weights to aggregate the RUL estimation results by several probabilistic support vector regression models. Their experimental validation demonstrated better prediction performance of the dynamic-weighted ensemble model than that of any single PSVR model. Although these aforementioned studies have investigated the effects of degradation on prognostic accuracy, very limited research has been conducted on how to best leverage these effects to improve the RUL prediction accuracy.

2.2. Prediction of engine performance

This section reviews the state-of-the-art in the prediction of RUL for aircraft engines. Chen et al. [27] presented a review on the RUL prediction of aircraft engines (or aeroengines) in 2011, where the applications of model-based, data-driven, and hybrid methods have been summarized. The PF, neural network and relevant vector machine were regarded as three promising techniques for RUL estimation. Recently, Alam et al. [28] integrated the ANN and autoregressive model into a progressive window framework to predict the RULs of aircraft turbine engines. Liu et al. [29] presented a Kalman filter-based degradation model that fused multiple monitoring parameters for aeroengine RUL estimation. In addition, a Bayesian degradation model for aeroengine RUL prediction was developed [30]. Lasheras et al. [31] proposed a data-driven model combining principal component analysis, dendrograms, and classification and regression trees for RUL prediction of aircraft engines. The principal component analysis and dendrograms were used to extract the most informative features from the multiple sensory data sets of the

aeroengine. Experimental analysis demonstrated that the feature extraction process improved the RUL prediction precision over the direct use of neural network model. Lim et al. [32] developed a data-driven model that integrated a feature-selection module and a time window neural network module to predict the RULs of aeroengines. Experimental results have shown that the K-means-based feature selection method can improve the prediction accuracy of the model trained by the ANN. Their analysis results demonstrated that the K-means based feature-selection benefitted the neural network prediction model. It is reasonable to get better prediction using feature-selection in the prediction process because the most informative features of the system degradation could be extracted from the sensory data sets. Bluvband and Porotsky [33] considered the suspended time-series of the NASA aircraft engine data sets. A series of support vector regression and support vector classification models were established to solve the missing data problem. The numerical analysis results indicate acceptable performance of their proposed method for aeroengine RUL prediction. Yuan et al. [34] employed long short-term memory neural network as a data-driven model to both diagnose engine faults and predict engine RULs. Yan et al. [35] proposed a data fusion method for monitoring and predicting the health status of an aircraft engine under different operation conditions. Specifically, a health index was introduced to measure the operation conditions of the engine using 21 sensors. This method accounted for the effects of different operation conditions in a unified manner. Experimental results have shown that the health index, derived from multi-sensor data, outperformed traditional methods that only use single sensor data. Xi et al. [47] proposed a copula-based sampling method for data-driven prognostics that predicts the RUL of a testing unit based on its degradation level. A copula-based statistical model was first constructed in the offline training phase to learn the statistical relationship between the failure time and the time realizations at specified degradation levels, and a simulation-based approach was used in the online testing phase to predict the probable failure time and RUL of a testing unit. The prognostic data sets of aircraft engines in the 2008 IEEE PHM Challenge Competition was employed to demonstrate the effectiveness of the proposed copula-based sampling method. Wang and Gao [36] presented a joint state and parameter estimation method for estimating the health of an aircraft engine. This method accounted for time-dependent degradation rates at different stages of engine operation by treating the operating parameters as time-varying variables. The prediction of engine performance was performed using PF under the Bayesian inference framework. A continuous resampling strategy was used to improve the prediction accuracy of PF. Experimental results have shown that PF is effective for detecting abrupt faults and predicting degradation of aircraft engines.

In summary, while a number of previous studies have investigated the effectiveness of ensemble learning methods in PHM, only a few papers have taken the effects of performance degradation into account, and to the best of our knowledge, none of these papers has predicted the performance degradation of aircraft engines using ensemble learning methods.

3. Methodology

3.1. A generic computational framework

A generic computational framework of the ensemble learning-based prognostic approach with degradation-dependent weights is illustrated in Fig. 1 and Table 1.

A training data set $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]^{\mathrm{T}}$ includes multi-dimensional measurement data from N different run-to-failure units, where $\mathbf{y}_i$ ($i = 1, 2, \ldots, N$) denotes the measurement data from the ith training unit. The training data set is used to train a predictive model. A test data set $\mathbf{y}_t$ denotes the measurement data from an online testing unit. The testing data set is used to validate the predictive model. A weight vector $\mathbf{w}^{s_t} = [w_1{}^{s_t}, w_2{}^{s_t}, \ldots, w_M{}^{s_t}]^{\mathrm{T}}$ denotes the weights associated with the degradation stage s_t of the testing unit, where M denotes the number of member algorithms. Because the weight-learning process recursively learns the health conditions of a system from new training data in an offline manner, the degradation-dependent weights are adaptively updated. The prediction accuracy of the ensemble prognostics can be improved by updating the weights associated with different degradation stages. More details about the offline training and online testing are presented in Sections 3.2 and 3.3.

3.2. Offline training phase

The objective of the offline training process, including degradation-stage classification, CV-testing, and weight-optimization, is to optimize the degradation-dependent weights associated with different degradation stages.

3.2.1. Classification of degradation stages

In degradation-stage classification, a number of degradation stages are predetermined for an engineered system. The degradation stages are defined based on the following guideline: The system units with similar degradation characteristics should be grouped into the same degradation stage. More specifically, the system units with health index ranging between certain values should be grouped into the same degradation stage. To determine the degradation stages, the locally weighted linear regression (LWR) method is introduced [37].

In LWR, a virtual health index (VHI) [38] is introduced as a data pre-processing scheme to transform the multi-dimensional measurement data set $\mathbf{Y}$ into one-dimensional VHI. Then, a smooth curve is constructed to fit the VHI index h and cycle index t using LWR such that the entire degradation process of each unit can be identified by the VHI values. Based on the VHI values, a number of different degradation stages can be predefined. It should be noted that there is a trade-off between the number of degradation stages and prediction accuracy. In general, the more degradation stages, the more accurate the predictive model is. However, more degradation stages will result in longer training time. In addition, because the original VHI includes noise, a smooth VHI-cycle curve will help avoid misclassification issues as well as balance the trade-off. LWR performs a regression around a point of interest on the VHI-cycle curve (i.e., a subset of the data points) instead of all the data points [37]. In this case, a fitted curve based on a subset of the data will not change due to the change of the data outside the subset.

The N training units are used to define the degradation stages since the damage in each training unit grows until the occurrence of at least a system failure. Based on the filtered VHI curves of the N training units, a stage-boundary index vector $\mathbf{B} = [b_0, b_1, \ldots, b_S]^{\mathrm{T}}$ can be determined, where b_0 is a constant VHI that determines the upper boundary of the 1st stage and b_s ($s = 1, 2, \ldots, S$ with S denoting the number of degradation stages) is a constant VHI value that determines the lower boundary of the sth stage. To determine the stage-boundary $\mathbf{B}$, an empirical model or an analytical model is required. For example, the empirical model typically requires expert domain knowledge on the system degradation features. The analytical model can be developed using clustering algorithms. In this work, for the purpose of alleviating the complexity of the proposed ensemble learning-based prognostics, the empirical model is employed to determine the stage-boundary $\mathbf{B}$. The expert domain knowledge on the degradation of aircraft engines is used to determine the boundary b_s that divides the training data set into moderate, severe and critical degradation stages.

3.2.2. Generation of CV-testing units

The RUL prediction of each member algorithm is generated through CV-testing on the CV-testing units in each degradation stage. More details about the CV-testing process is presented as follows:

Step 1 : Generate the synthetic CV-testing units for each degradation stage. These CV-testing units are conducted by truncating the run-to-failure VHI data of each training unit in $\mathbf{Y}$ for S times (i.e., for S stages)

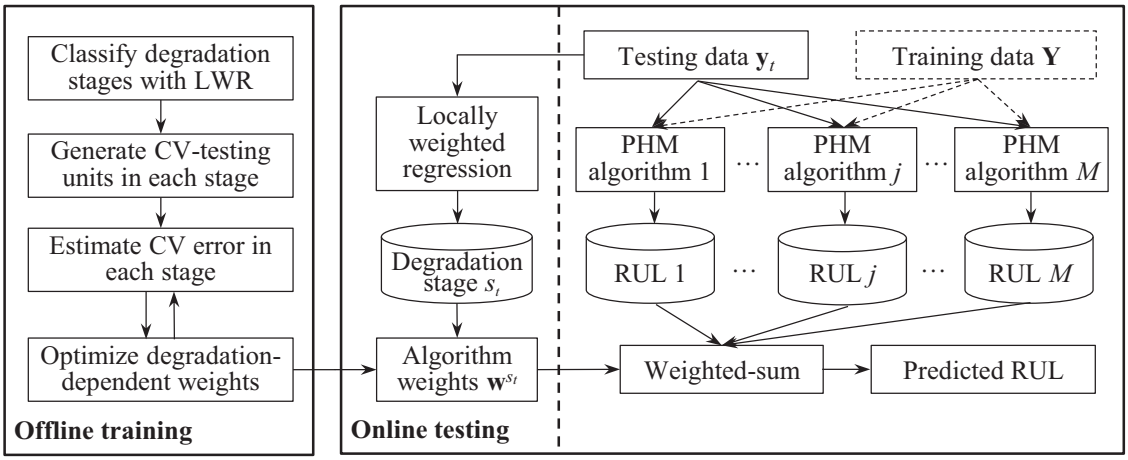

Fig. 1. The flowchart of the proposed method for ensemble prognostics.

Table 1
A generic computational framework.

1.	Offline training: Compute the degradation-dependent weight vector for each degradation stage (Section 3.2)
1.1	Determine the degradation stages using the locally weighted linear regression method (Section 3.2.1)
1.2	Generate CV-testing units for each degradation stage by random truncations (Section 3.2.2, Step 1)
1.3	Perform CV-testing for each degradation stage (Section 3.2.2, Step 2)
1.4	Optimize the weight vector in each degradation stage (Section 3.2.3)
2.	Online testing: Make predictions using the degradation-dependent weights (Section 3.3)
2.1	Identify the degradation stage of the online testing unit (Section 3.3.1)
2.2	Predict the RULs of the testing unit using M member algorithms (Section 3.3.2)
2.3	Make predictions using the predictive model trained by the ensemble learning-based prognostic approach (Section 3.3.2)

at the cycles corresponding to S pre-assigned RULs [12]. The RUL that is pre-assigned to each training unit for the sth stage is randomly generated from a uniform distribution between the minimum and maximum values of the RULs of the unit. The range of the uniform distribution for a degradation stage is selected based on the fact that the variation of the pre-assigned RULs should be sufficiently great to represent the performance of individual algorithms at the stage. For example, it is assumed that the filtered VHI curves of N^s training units partially fall into the VHI range of the sth ($s=1, 2,\ldots, S$) degradation stage. For the ith ($i=1, 2,\ldots, N_s$) training unit, this step first identifies the minimum and maximum RULs, $L_i^{s,\min}$ and $L_i^{s,\max}$, of the unit if the fitted VHI of the unit falls into the sth stage, then randomly selects an RUL, $L_i^{s_T}$, that falls between $L_i^{s,\min}$ and $L_i^{s,\max}$, and truncates the VHI data of the training unit at the cycle corresponding to $L_i^{s_T}$. Finally, we obtain N^s synthetic CV-testing units for the sth degradation stage whose pre-assigned RULs are $\mathbf{L}_s{}^T = [L_1{}^{s_T}, L_2{}^{s_T},\ldots, L_{Ns}{}^{s_T}]$. $\mathbf{I}^s = \{i \in [1, N] | s \in \mathbf{s}(\mathbf{y}_i)\}$ denotes an index set that contains the indices of all training units that are used to generated the N^s CV-testing units. $\mathbf{s}(\mathbf{y}_i)$ denotes a set of degradation stages where the filtered VHI data of the ith training unit $\mathbf{y}_i$ lie. Ideally, each training unit can be used to generate CV-testing units for all degradation stages (i.e., $N^s = N$, for $s=1, 2,\ldots, S$); however, since the training units may start from different initial health conditions, the initial VHI h_{i0} may fall below the boundaries of the first several stages (e.g., b_1, b_2 and b_3). Therefore, the numbers of CV-testing units for the first several stages may be less than N (i.e., $N^s < N$).

Fig. 2 illustrates the generation of 3 CV-testing units from a training unit (e.g., unit #1), where three degradation stages ($S=3$) are considered. The time-series VHI data of the training unit are truncated at three cycles corresponding to three randomly selected RULs (i.e., $L_1{}^{1T}$, $L_1{}^{2T}$ and $L_1{}^{3T}$). As a result, 3 CV-testing units with pre-assigned RULs (i.e., $L_1{}^{1T}$, $L_1{}^{2T}$ and $L_1{}^{3T}$) are generated, each of which is associated with a specific degradation stage.

Step 2 : Conduct k-fold CV with the CV-testing units for each degradation stage. This step randomly divides the CV-testing data set for each stage into k disjoint subsets. The size of each subset is approximately the same [12]. Let $\mathbf{I}_s^m$, $m=1, 2,\ldots, k$, denote the index set of the training units whose measurement data generate the CV-testing units in the mth subset for the sth stage. It easily follows that $\mathbf{I}^s = \cup_{m=1}^k \mathbf{I}_m^s$ and $\sum_{m=1}^k N_m^s = N^s$. The CV process is conducted k times. In each CV trial, the CV-testing units from the training units in one of the k subsets are

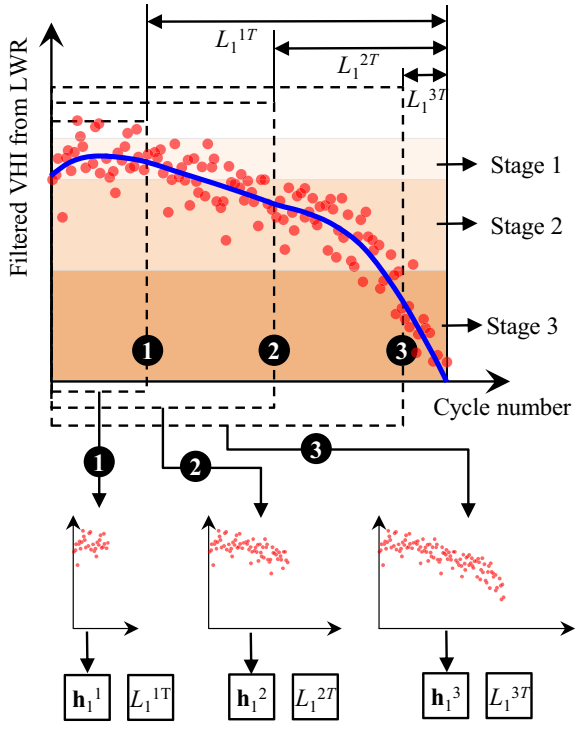

Fig. 2. Generation of 3 CV-testing units from a training unit (unit #1) via truncating the VHI data of the training unit.

used for testing; the training units in the other $k-1$ subsets are used for training. Thus, each CV-testing unit from $\mathbf{I}_s$ is used once for testing, and each training unit in $\mathbf{I}_s$ is used $k-1$ times for training.

During the k-fold CV, the M member algorithms predict RULs, $\hat{\mathbf{L}}^s = [\hat{\mathbf{L}}_1^s, \hat{\mathbf{L}}_2^s, \ldots, \hat{\mathbf{L}}_M^s]^{\mathrm{T}}$, where $\hat{\mathbf{L}}_j^s$ denotes the predicted RULs of the CV-testing units by the jth member algorithm. The CV error for the sth degradation stage is computed by taking an average across all k trials as expressed below:

$$\varepsilon_{CV}^s = \frac{1}{N^s}\sum_{m=1}^{k}\sum_{i\in\mathbf{I}_m^s} e\left(\hat{L}_i^s\left(\mathbf{w}^s, \hat{\mathbf{L}}^s\left(\mathbf{y}_i^s, \mathbf{Y}\backslash\mathbf{Y}_m\right)\right), L_i^{sT}\right) \tag{1}$$

where $\hat{L}_i^s$ denotes the ensemble-predicted RUL of the ith CV-testing unit at the sth stage; $e(\bullet)$ is a predefined evaluation metric that measures the accuracy of the ensemble prediction; $\mathbf{w}^s = [w_1{}^s, w_2{}^s, \ldots, w_M{}^s]^{\mathrm{T}}$ is the weight vector of the member algorithms associated with the sth stage; $\mathbf{y}_i{}^s$ is the measurement data from the CV-testing unit; $\mathbf{Y}_m$ is the mth training data subset; and $L_i{}^{sT}$ denotes the true RUL of the ith unit at the sth stage. The outcome of Step 2 is the CV error for each degradation stage.

3.2.3. Optimization of degradation-dependent weights

To reduce the CV error, the degradation-dependent weight vectors $\mathbf{w} = [\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^S]^{\mathrm{T}}$ associated with the S degradation stages need to be optimized. In the previous research [12], an optimization-based weighting scheme was proposed to maximize the accuracy and robustness of an ensemble by synthesizing the prediction accuracy and diversity of its member algorithms. In this study, the optimization-based weighting scheme is used to optimize the degradation-dependent weight vectors. The weights for the sth degradation stage can be determined by solving the following optimization problem:

$$\begin{aligned} &\underset{\mathbf{w}^s}{\text{Minimize}} \quad \varepsilon_{CV}^s = \varepsilon_{CV}^s\left(\hat{L}_i^s\left(\mathbf{w}^s, \hat{\mathbf{L}}^s\left(\mathbf{y}_i\right)\right), L_i^{sT}, i \in \mathbf{I}^s\right) \\ &\text{Subject to} \quad \sum_{j=1}^{M} w_j^s = 1 \end{aligned} \tag{2}$$

where $\mathbf{I}^s$ is an index set that contains the indices of all training units whose degradation stages are s. It is expected that the resulting ensemble with optimized degradation-dependent weights will optimally combine the generalization capabilities of the member algorithms and achieve robust RUL prediction. Given a pool of a large number of prognostic algorithms, solving the weight optimization problem in Eq. (2) would allow an optimal selection of prognostic algorithms from the pool to be used in the online testing phase. This is because larger weights associated with a degradation stage are likely to be assigned to algorithms that produce higher prognostic accuracy and diversity in the stage, while near-zero weights associated with the degradation stages are likely to be assigned to algorithms that perform significantly more poorly than other algorithms in the stage.

3.3. Online testing phase

3.3.1. Identification of degradation stages for online testing units

Upon the determination of the stage-boundary $\mathbf{B}$ in the offline phase (see Section 3.2.1), the degradation stage of an online testing unit $\mathbf{y}_t$ can be identified based on the most recent filtered VHI value. Specifically, LWR is first performed on the VHI data of $\mathbf{y}_t$ to generate a filtered VHI-cycle curve; then, the filtered VHI value at the current cycle is compared with $\mathbf{B}$ to determine the degradation stage that $\mathbf{y}_t$ belongs to. Once the degradation stage s_t is determined for $\mathbf{y}_t$, the weights $\mathbf{w}^{s_t}$ associated with s_t can be used to produce an ensemble-predicted RUL with a weighted-sum formulation.

It is worth distinguishing between operation conditions and degradation stages. Operation conditions are a set of conditions (e.g., temperature, pressure, and current) under which an engineered system operates; a degradation stage is defined as a period of time when the health condition of an engineered system stays in a certain range (i.e., in terms of the filtered VHI value). Operation conditions often affect the degradation rate of a system and thus how fast the system transitions from one degradation stage to another. As long as the individual prognostic algorithms in an ensemble can handle static and varying operation conditions in RUL prediction, the proposed ensemble learning-based prognostic approach should also be capable of dealing with time-dependent degradation due to both static and varying operation conditions. It should be noted, however, that the focus of this paper is to accurately predict the RUL of an engineered system by accounting for the effects of time-dependent degradation rather than those of varying operation conditions.

3.3.2. Formulation of ensemble prognostics with degradation-dependent weights

The predicted RULs of an online testing unit $\mathbf{y}_t$ by M member algorithms are aggregated to generate the ensemble- predicted RUL for the testing unit using the following weighted-sum formulation [12]:

$$\hat{L} = \sum_{j=1}^{M} w_j^{s_t}\hat{L}_j\left(\mathbf{y}_t, \mathbf{Y}\right) \tag{3}$$

where $\mathbf{y}_t$ denotes the measurement data from the online testing unit; $\hat{L}$ denotes the ensemble-predicted RUL for $\mathbf{y}_t$; $w_j{}^{s_t}$ denotes the weight assigned to the jth prognostic algorithm associated with the degradation stage s_t; $\hat{L}_j(\mathbf{y}_t, \mathbf{Y})$ denotes the predicted RUL by the jth prognostic member algorithm trained with the data set $\mathbf{Y}$. Let the weight vector $\mathbf{w}^{s_t} = [w_1{}^{s_t}, w_2{}^{s_t}, \ldots, w_M{}^{s_t}]^{\mathrm{T}}$ and the vector of predicted RULs by member algorithms $\hat{\mathbf{L}} = [\hat{L}_1, \hat{L}_2, \ldots, \hat{L}_M]^{\mathrm{T}}$, the weighted-sum formulation in Eq. (3) can be expressed in a vector form as $\hat{L}(\mathbf{w}^{s_t}, \hat{\mathbf{L}}) = (\mathbf{w}^{s_t})^{\mathrm{T}}\hat{\mathbf{L}}$.

It is noted that the computationally expensive process of training multiple prognostic algorithms is carried out in the offline phase (see Section 3.2) and the process of predicting the RUL of an online testing unit with the multiple trained algorithms (see Eq. (3)) requires a relatively small amount of computational effort. Therefore, the proposed ensemble prognostics method raises little concern for the computational complexity. Indeed, in many practical applications, prognostic accuracy is often more important than computational complexity. This is because the catastrophic failure of an engineered system often costs much more than the effort to increase the computational power in the online phase. It then follows that, in cases where an ensemble of prognostic algorithms produces a considerably more accurate prediction of RUL over any of its members, ensemble prognostics should always be preferred over the use of any standalone algorithm.

4. Case 1: RUL prediction for aeroengine bearings

In the first case study, the ensemble learning-based prognostic approach with degradation-dependent weights is used to predict the RUL of an aeroengine bearing. The degradation data of the aeroengine bearings were generated from a crack-growth simulation model presented in [44]. The proposed ensemble prognostics method with degradation-dependent weights (EDD) is demonstrated using the crack-degradation data and compared with the original ensemble prognostics method with degradation-independent weights (EDI).

4.1. Description of bearing crack-growth data

The phase space warping (PSW) [45], which uses short-time reference model prediction error to measure small deteriorations in the phase space of a fast-time subsystem, can be used to develop a linear relationship between a damage tracking metric (e.g., the change in the direction of a phase space trajectory) and the length of a crack [44], expressed as

$$x = \alpha z \tag{4}$$

where x is the damage tracking metric estimated by PSW, z is the crack length, and α is a constant coefficient. As a result, the tracking metric

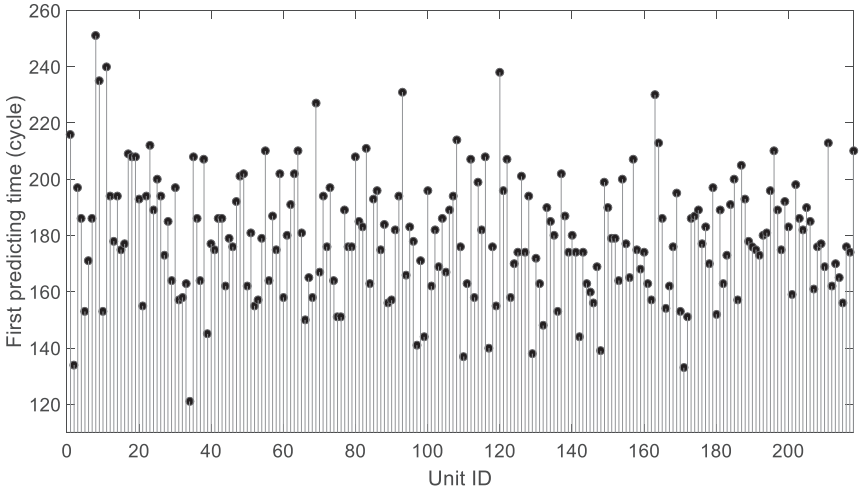

Fig. 3. The distribution of the FPT values of the 218 training units.

x can be used as the health indicator (or health index) for RUL estimation. Considering that the tracking metric x is proportional to the crack length, and in theory the crack propagation often consists of one slow-growth and one rapid-growth degradation stages, a simulation model [44] can be established to describe the propagation of x in an aero-engine bearing as

$$x = \begin{cases} (0.002t + 1)/10 + 0.01 * \lambda, & t \le \kappa \\ \beta^t/10/\gamma + 0.01 * \lambda, & t > \kappa \end{cases} \quad (5)$$

where t denotes the operation cycles of the bearing; λ is a normally distributed random number, describing the measurement noise; κ denotes the first predicting time (FPT) that defines the changing point of the bearing degradation stages; β and γ are constant coefficients. The simulation model in Eq. (5) represents the damage prorogation characteristics at the slow-growth stage ($t \le \kappa$) and rapid-growth ($t > \kappa$) stage of a bearing.

In [44], a set of parameters was selected to simulate the tracking metric x of a bearing unit ($\beta = 1.04$, $\gamma = 1821.96$ and $\kappa = 200$). In this paper, the simulation data collected from bearing units with diverse propagation characteristics of tracking metrics were used to demonstrate the ensemble learning-based prognostic approach. Note that in Eq. (5) the FPT determines the time when the damage propagation process of a bearing unit falls into the rapid-growth degradation stage. Different FPT values result in dramatically different crack propagation characteristics. Hence, 218 randomly generated FPTs (see Fig. 3) were used to produce degradation data for 218 training bearing unis. The values of FPT range from 121 to 251 cycles. Moreover, the model coefficients γ for the 218 bearing units were randomly generated from a uniform distribution between 1472 and 2172. Fig. 4(a) depicts the tracking metric curves of four training bearing units. As shown in Fig. 4(a), the degradation processes of the four units are significantly different due to their different FPTs. Fig. 4(b) shows the tracking metric curves of all 218 training bearing units. The red dots in Fig. 4(b) show the tracking metrics of the 218 training units, plotted against the adjusted cycle number that is defined as the subtraction of the cycle-to-failure of a training unit from the actual operational cycle of the unit. The blue curves in Fig. 4(b) represent the LWR-fitted curves for the 218 training bearing units.

In addition to the training data sets, another 218 testing bearing units were simulated using Eq. (5) with similar parameter settings as the training units, and the tracking metric curve of each testing unit was truncated at some cycle prior to the end of life (EOL) cycle when a system failure occurred (i.e., failure threshold was 1.0 in this case).

4.2. Optimization of degradation-dependent weights

The proposed ensemble prognostics method described in Section 3 is demonstrated using the bearing data sets. Two prognostic algorithms were selected as the member algorithms of the ensemble prognostics method. One was a data-driven algorithm, i.e., the similarity-based interpolation (SBI) approach [38] with the relevance vector machine (RVM) [40] (RS); and the other one was a model-based algorithm, i.e., the particle filter (PF) [46]. Based on our previous study on ensemble prognostics [12], a 10-fold CV was used in the optimization of degradation-dependent weights for each degradation stage. The 10-fold CV involved partitioning the training data set into 10 mutually exclusive subsets, training a predictive model on nine subsets (or CV-training set), and validating the predictive model on the remaining subset (or CV-testing set). The process of generating synthetic CV-testing units in a CV-testing set is detailed in steps 1–4.

Step 1 : Define the degradation stages (see Section 3.2.1). It can be seen that the LWR-fitted curves in Fig. 4(b) can be divided into two degradation stages, separately by a constant tracking metric value (0.145). These two stages correspond to the slow-growth and rapid-growth degradation stages of the bearing cracks.

Step 2 : Generate partial degradation data via truncations of run-to-failure training bearing data sets (see Section 3.2.2, Step 1). Since there were 218 training units, we truncated each unit once in each degradation stage to generate 218 CV-testing units in each stage. The full run-to-failure tracking metric curve of each training unit was truncated at a pre-assigned RUL that was randomly chosen within the full cycles of that training unit. For example, assume that the EOL cycle of a training unit is 200. The pre-assigned RUL for this unit is then randomly generated from a uniform distribution on the RUL interval [0, 200]. Let us further assume this pre-assigned RUL takes a value of 150. Then, the cycle number for the pre-assigned RUL can be computed by subtracting the RUL from the EOL cycle as 50 (= 200 − 150). As a result, a CV-testing unit with a pre-assigned RUL (150 cycles) and 50 cycles of tracking metric data is generated from this training unit. In this case study, the numbers of the CV-testing units whose LWR-fitted curves partially fall into the 2 stages are 218 and 218, respectively.

Step 3 : Perform the 10-fold CV on the 218 CV-testing units in each stage (see Section 3.2.2, Step 2). Two prognostic algorithms (i.e., RS and PF) were used to predict the RUL of each CV-testing unit. For instance, in the first CV trial in stage 1, the complete (run-to-failure) tracking metric measurements from the first 198 units (i.e., subsets 1–9) were used as the training data, and the partial tracking metric measurements truncated from the remaining 10 (CV-testing) units as the testing data. So in the 10 CV trials, each of the 218 CV-testing units in this stage was tested by once.

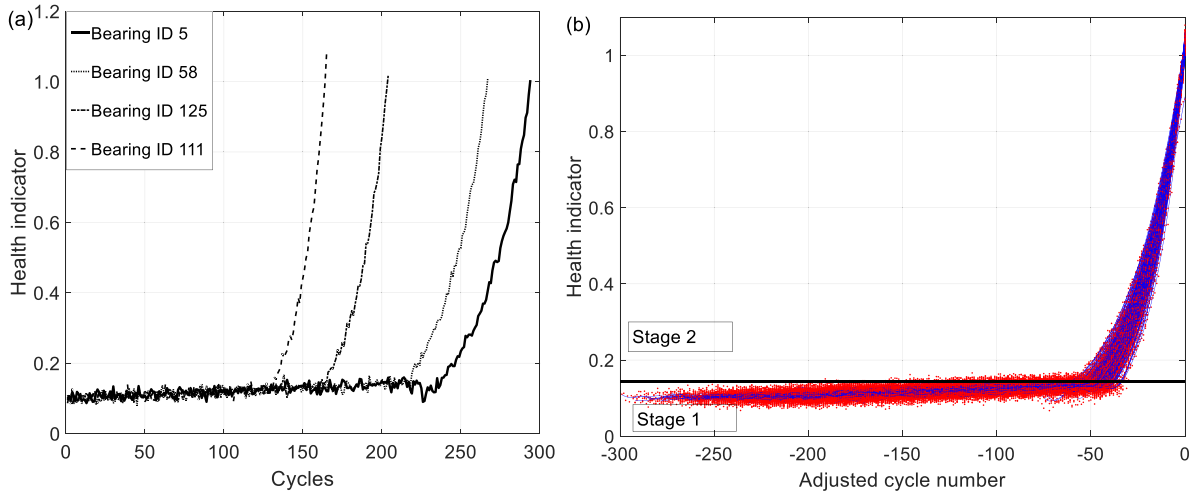

Fig. 4. The tracking metric curves of (a) 4 bearing units and (b) 218 bearing units.

Table 2
Degradation-dependent ($\mathbf{w}^{1-2}$) and degradation-independent ($\mathbf{w}$) weights.

Weight vector	RS	PF
$\mathbf{w}^1$	0.5950	0.4050
$\mathbf{w}^2$	0.0239	0.9761
$\mathbf{w}$	0.5876	0.4124

Step 4 : Optimize the degradation-dependent weights by minimizing the CV error for each stage (see Section 3.2.3). The CV error used in this case study is measured with an asymmetric score function around the true RUL such that greater penalties are applied on late predictions than early ones [12,39]. The score evaluation metric ε_{SC} can be expressed as

$$\varepsilon_{SC}\left(\hat{L}_i, L_i^T\right) = \begin{cases} \exp\left(-d_i/13\right) - 1, & d_i < 0 \\ \exp\left(d_i/10\right) - 1, & d_i \geq 0 \end{cases} \quad \text{where } d_i = \hat{L}_i - L_i^T \quad (6)$$

where $\hat{L}_i$ and $L_i{}^T$ denote the predicted and true RULs of the ith CV-testing unit, respectively. This score function was used to compute, for each degradation stage, the CV errors by the member algorithms and that by an ensemble model (see $\varepsilon_{CV}{}^s$ in Eq. (1)) for a given set of degradation-dependent weights. For stage 1, the RUL predictions of the 104 CV-testing units by RS and PF were used to evaluate the CV error in the stage, and the weight optimization was implemented to identify an optimum set of algorithm weights that minimized the CV error. In this study, the weight optimization problem in Eq. (2) was solved using a sequential quadratic optimization (SQP) method, which is a gradient-based optimization technique. We then repeated this procedure to obtain the optimized weights for stage 2. We finally obtained 2 weight vectors $\mathbf{w}^{1\text{-}2}$ for stages 1 and 2, respectively. Table 2 summarizes the degradation-dependent weights ($\mathbf{w}^{1\text{-}2}$) and degradation-independent weights ($\mathbf{w}$), where the weight vector $\mathbf{w}$ was calculated by minimizing the overall CV error of the 436 (= 218 + 218) CV-testing units.

4.3. RUL prediction results

4.3.1. Ensemble prediction on CV-Testing units

After calculating the degradation-dependent weight vectors, the ensemble prediction result of each stage can be calculated using the weighted-sum formulation in Eq. (3). Table 3 compares the RUL prediction errors on the initial 436 CV-testing units by EDI and EDD. It can be seen from the table that the prediction error generated by EDD was smaller than that of EDI at both stages 1 and 2, and the prediction improvement in stage 2 was 58.50%. This resulted in a 1.35% reduction in the overall CV error that was computed with all the 436 CV-testing units. Fig. 5 manifests the predictions of RS, PF, EDI and EDD on the initial CV-testing units. The units are sorted by their RULs in an ascending order. As shown in Fig. 5, the prediction accuracy was significantly improved by the two ensemble learning-based prognostic approaches (EDI and EDD) when compared with PF and RS (whose overall CV errors were 99.5910 and 22.6437, respectively). More importantly, EDD outperforms EDI in stage 2 (see Fig. 5(b)).

In order to validate the performance improvement by the proposed ensemble prognostics method, we independently generated another set of synthetic CV-testing units (i.e., repeated 436 CV-testing units) by randomly truncating the tracking metric curves of the 218 training units. The 10 CV trials were repeated with repeated CV-testing, and the weight vectors ($\mathbf{w}^{1\text{-}2}$ and $\mathbf{w}$) obtained from the initial CV (see Table 2) were used to compute the CV errors by EDI and EDD. Fig. 6 shows the prediction of RUL on the CV-testing units using RS, PF, EDI and EDD methods. As shown in Table 3, the proposed EDD method outperforms the original EDI method on the repeated CV-testing units. The CV errors of EDD were reduced in stages 1 and 2 against EDI. It should also be noted that the prediction errors by both EDD and EDI in stage 2 are much smaller than those in stage 1 (see Table 3 and Figs. 5(b) and 6(b)). This can be attributed to two performance factors associated with RS and PF (i.e., the two member algorithms of EDD and EDI). First, RS, as a data-driven method, is generally more accurate given a larger number of measurements in CV-testing data because more measurements often provide more important/accessible information about the distinct characteristics of the degradation. Thus, the prediction accuracy of RS is relatively low in stage 1 (see Figs. 5(a) and 6(a)), where the tracking metric measurements from a CV-testing unit are limited. As the measurement length of CV-testing data increases in stage 2, the prediction accuracy of RS also increases. Second, PF, as a model-based method, is typically more effective when making predictions for a relatively short time frame ahead. In stage 1, PF is not as effective as that in stage 2 because the time frame in which predictions need to be made in stage 1 is longer than stage 2. Since both RS and PF make better predictions in stage 2 than stage 1, the prediction results of the ensemble prognostics methods (EDD and EDI) are also more accurate in stage 2 than stage 1.

It should be noted that in Fig 6 the overall CV errors of RS and PF were 18.6050 and 74.2787, respectively. This result is consistent with the observations from Table 3 and Fig. 5. In addition, the ensemble prognostics methods (EDI and EDD) outperform the individual member algorithms as expected. Moreover, as shown in Table 3 and Figs. 5 and 6, the performance of EDD is similar to that of EDI in stage 1, while EDD outperforms EDI in stage 2. This is because EDI is designed to maximize the overall prediction accuracy (over both degradation stages) by opti-

Table 3
CV-testing errors by the original (EDI) and proposed (EDD) ensemble prognostics methods.

Degradation stage	Range of current tracking metric	Initial CV-testing			Repeated CV-testing		
		Units No.	*S*/EDI	*S*/EDD	Units No.	*S*/EDI	*S*/EDD
1	[0, 0.145]	218	25.0754	25.0685	218	24.5524	24.1371
2	[0.145, 1]	218	1.4758	0.6124	218	1.4708	1.1255
Overall	[0, 1]	436	13.2756	13.0970	436	13.0563	12.8040

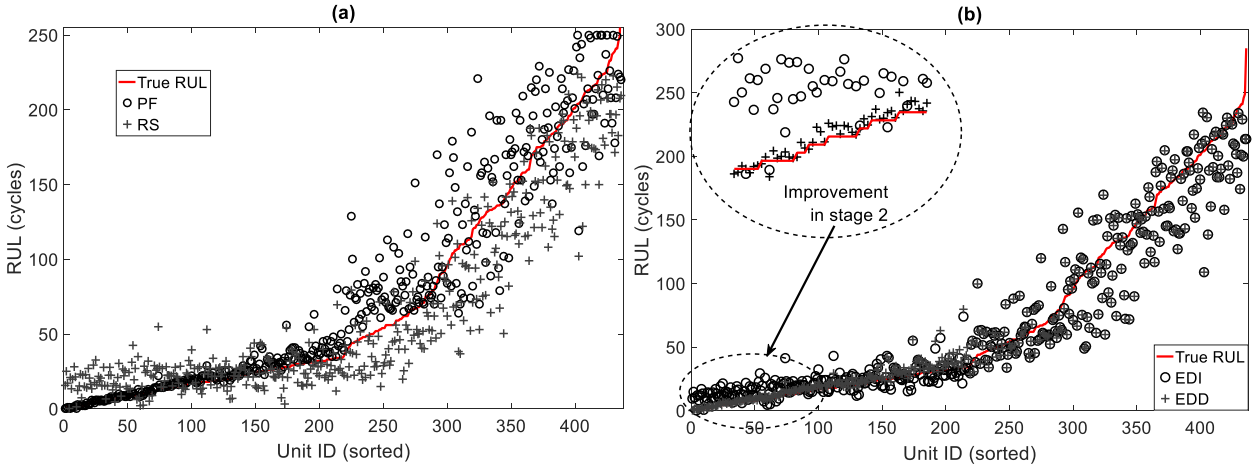

Fig. 5. RUL predictions on initial CV-testing units: (a) member algorithms, and (b) ensemble learning-based methods.

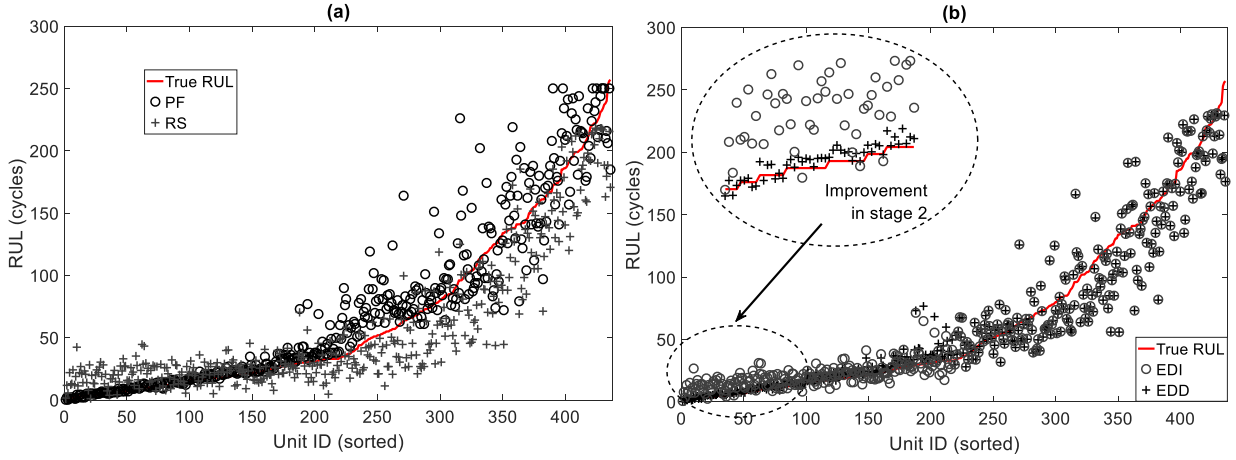

Fig. 6. RUL predictions on repeated CV-testing units: (a) member algorithms, and (b) ensemble learning-based methods.

mizing the weights of RS and PF. As shown in Figs. 5 and 6, because the prediction accuracy of both RS and PF was much worse in stage 1 than stage 2, in the weight-optimization process EDI has made much larger effort in obtaining satisfactory prediction performance in stage 1 than it did in stage 2 in order to maximize the overall prediction accuracy. As a result, the weight vector **w** was not optimum for stage 2. On the other hand, EDD adopted a local optimization strategy for individual degradation stages such that the optimal prediction performance can be achieved for each stage.

4.3.2. *Testing data set for performance validation*

In order to evaluate the robustness of the proposed ensemble prognostics method (EDD), the 218 testing bearing units were used to examine its prediction performance. The validation process is summarized in Table 4. Table 5 and Fig. 7 depict the validation results on the 218 testing data sets. The units are sorted by the RULs in an ascending order in Fig. 7. In the validation, there were 118 and 100 units in stages 1 and 2, respectively. As can be seen in Table 5, the prediction error of EDD

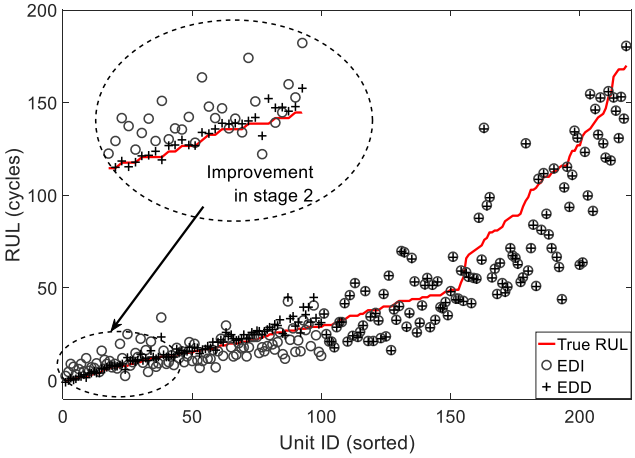

Fig. 7. RUL predictions on validation units using EDD and EDI methods.

Table 4
Procedure of the validation process.

Step 1	Classify the degradation stage of each testing units into the two stages based on the filtered tracking metric curve.
Step 2	Perform the 10-fold CV to obtain the estimated RULs of the 218 testing units using RS and PF. The prediction matrixes $\hat{\mathbf{L}}^s$ ($s = 1$ and 2), correspond to the two degradation stages, respectively.
Step 3	Implement the weighted-sum formulation (Eq. (3)). By multiplying the weight vector $\mathbf{w}^s$ in Table 2 with $\hat{\mathbf{L}}^s$ ($s = 1, 2,$ and 3), the ensemble prediction for each degradation stage can be obtained.
Step 4	Calculate validation errors of the proposed ensemble prognostics method (EDD).
Step 5	Compare the proposed ensemble (EDD) with the original ensemble (EDI). The weight vector $\mathbf{w}$ in Table 2 is used for the EDI ensemble.

Table 5
Prediction errors in the validation.

Degradation stage	Range of current tracking metric	No. of CV-testing units	Prediction error (score function)			
			RS	PF	EDI	EDD
1	[0, 0.145]	118	34.8475	557.2365	13.9044	13.8592
2	[0.145, 1]	100	1.7425	0.3908	0.7717	0.3787
Overall	[0, 1]	218	19.6617	301.8027	7.8802	7.6755

in stage 1 is comparable to EDI prediction results while in stage 2 and overall are smaller than that of EDI with the improvements of 50.93%, and 2.60%, respectively. The improvement in stage 2 was larger than that in stage 1, which was consistent with that obtained in the CV-testing. In addition, both the EDD and EDI ensemble methods improved the prediction precision dramatically over the individual member algorithms. These observations are consistent with the prediction results in Table 3, which suggests strong robustness ability of the proposed ensemble (EDD).

5. Case 2: RUL prediction for aircraft engines

5.1. Description of engine performance simulaton data

In the second case study, the ensemble learning-based prognostic approach with degradation-dependent weights is used to predict the RUL of an aircraft engine. The prognostic data sets provided for the 2008 IEEE PHM Data Challenge Competition consist of multivariate time series signals that are collected from engine dynamic simulations [39]. The objective of this data challenge competition was to predict the RUL of an aircraft engine that operates under six different flight conditions. The data sets were generated by an aero-propulsion system simulator, C-MAPSS, for developing, testing, and validating data-driven prognostic algorithms. The scenario developed for the challenge data tracks 536 aircraft engine units throughout their usage history. The prognostic data at each operation cycle of each engine unit include the unit ID, cycle index, the values of 3 operation-condition parameters (i.e., altitude, match number, sea-level temperature), and the values of 21 sensor measurements. The 3 operation-condition parameters have substantial effects on the engine performance and degradation, and an engine unit may operate under varying operation (or flight) conditions that fall within six unique combinations of the operation-condition parameters [39]. The sensor measurements typically include noise. The main sources of the noise are manufacturing and assembly variations, process noise, and measurement noise. More details about how the noise is modeled can be found in [39]. In the case study, 536 data sets were divided into training and testing data sets, each with 218 data sets. In the training data set, each engine unit ran from its initial health condition to a system failure; and in the testing data set, the time series signals of each unit were truncated at some cycle prior to the EOL cycle when a system failure occurs.

5.2. Fundamentals and implementations of member algorithms

Five prognostic algorithms were selected as member algorithms in the ensemble. These five algorithms include:

1. The SBI with RVM (RS) [38,40];
2. SBI with SVM (SS) [38,41];
3. SBI with the least-square exponential fitting (ES) [38];
4. Bayesian linear regression with the least-square quadratic fitting (QB) [42]; and
5. Recurrent neural network (RNN) [43].

The virtual health index (VHI) [38] was used as a data pre-processing scheme for the first four algorithms, while a simple normalization scheme is used for the last algorithm. The selection of the member algorithms and the use of the VHI were motivated by the previous work on ensemble prognostics [12]. Based on the results in [12,38], seven most informative sensory signals among the 21 sensor signals were selected to construct a transformation matrix $\mathbf{T}$ and compute the VHI [38] when using the first four member algorithms (RS, ES, SS and QB). The construction of the virtual health index (VHI) explicitly considers the varying operation conditions (or flight) by taking the condition index as an additional input during data pre-processing. RNN uses 21 normalized signals. More details on the parameter settings for these five member algorithms can be found in [12].

5.3. Optimization of degradation-dependent weights

The optimization of the degradation-dependent weights expressed in Eq. (2) requires the evaluation of the CV error for each degradation stage. The process of generating synthetic CV-testing units is detailed in Steps 1–4.

Step 1 : Calculate the VHI values of the 218 training units. A linear data transformation method with the matrix $\mathbf{T}$ was used to transform the multi-dimensional sensor signals to one-dimensional VHI [38]. The red dots in Fig. 8 show the calculated VHI data of the 218 training units, plotted against the adjusted cycle number that is defined as the subtraction of the cycle-to-failure of a training unit from the actual operational cycle of the unit.

Step 2 : Perform LWR on the VHI data for each of the 218 training units to obtain their fitted VHI curves (see Section 3.2.1). The blue curves in Fig. 8 represent the VHI curves for the 218 training units.

Step 3 : Define the degradation stages (see Section 3.2.1). In this example, the VHI range was divided into 3 stages: [0.7, 1.2] (stage 1), [0.4, 0.7] (stage 2), and [−0.2, 0.4] (stage 3). In stage 1, the training units are relatively healthy and generally exhibit low degradation rates. In stage 2, the training units deteriorate more rapidly, and exhibit severe degradation. In stage 3, the degradation of the training units continues to accelerate, meaning that these units tend to fail. The boundaries between these three stages are shown in Fig. 8 using the black horizontal lines. In this case study, the numbers of the training units whose filtered VHI curves partially fall into the 3 stages are 127, 202 and 218, respectively.

Table 6
CV-testing errors by member algorithms for each degradation stage.

Degradation stage	Range of current VHI	No. of CV-testing units	CV-testing error (score function)				
			RS	ES	SS	QB	RNN
1	[0.7, 1.2]	127	27.1590	22.0642	23.2310	42.4690	63.8269
2	[0.4, 0.7]	202	10.0675	8.1175	8.7331	718.3503	63.5937
3	[−0.2, 0.4]	218	1.8242	1.5362	1.5739	124.3239	2.1903

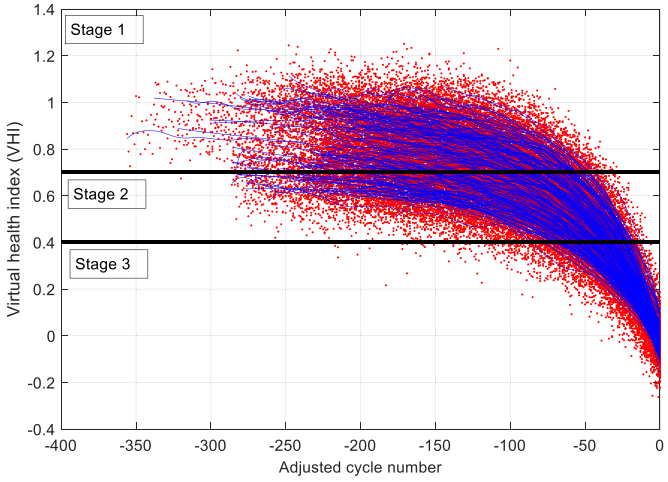

Fig. 8. VHI curves from locally weighted regressions on the VHI data of the 218 training units and the degradation stage classification.

Step 4 : Generate partial degradation data via truncations of run-to-failure VHI data (see Section 3.2.2). These partial degradation data, along with the true RULs, constituted the CV-testing units. The generation of the partial degradation data was performed for each degradation stage by truncating the full run-to-failure VHI data at pre-assigned RULs. For example, assume that the VHI data of a training unit that falls into stage 1 corresponds to a minimum RUL of 100 cycles and a maximum RUL of 250 cycles (i.e., the EOL cycle of the unit). The pre-assigned RUL for this unit is then randomly generated from a uniform distribution on the RUL interval [100, 250]. Let us further assume this pre-assigned RUL takes a value of 180. Then, the cycle number for the pre-assigned RUL can be computed by subtracting the RUL from the EOL cycle as 70 (= 250 − 180). As a result, a CV-testing unit with a pre-assigned RUL (180 cycles) and 70 cycles of VHI data is generated for stage 1. Since one CV-testing unit was generated from one training unit for each stage, the number of the CV-testing units for each stage was the same as that of the training units whose VHI data partially fall into the stage.

Once the CV-testing units are generated, the next step is to determine the degradation-dependent weights by minimizing the CV error for each stage (see Section 3.2.3). In this case study the 10-fold CV is used to evaluate the prediction error. In the 10-fold CV process, five prognostic algorithms (i.e., RS, ES, SS, QB, and RNN) were used to predict the RUL of each CV-testing unit in each stage. The score evaluation metric ε_{SC} in Eq. (6) was used here to measure the CV errors. Table 6 summarizes the CV errors produced by the member algorithms for each degradation stage.

Once the CV-testing errors are computed, degradation-dependent weights need to be optimized to minimize the error in each stage. For example, in stage 1, the RUL predictions of the 127 CV-testing units by the member algorithms (i.e., 127 predicted RULs from each algorithm) were used to evaluate the CV error in the stage, and the weight optimization was implemented to identify an optimum set of algorithm weights that minimized the CV error. The weight optimization problem in Eq. (2) was then solved using SQP. We then repeated this procedure to obtain the optimized weights for stages 2 and 3. We finally obtained 3 wt vectors $\mathbf{w}^{1\text{-}3}$ for stages 1–3, respectively. Table 7 summarizes the degradation-dependent weights ($\mathbf{w}^{1\text{-}3}$) and degradation-independent weights ($\mathbf{w}$), where the weight vector $\mathbf{w}$ was obtained by minimizing the overall CV error considering the CV-testing unis from all three stages (i.e., 127 + 202 + 218 = 547 CV-testing units).

Table 7
Degradation-dependent ($\mathbf{w}_{1\text{-}3}$) and degradation-independent ($\mathbf{w}$) weights of member algorithms obtained from optimization-based weighting.

Weight vector	RS	ES	SS	QB	RNN
$\mathbf{w}^1$	0.0000	0.7011	0.0000	0.2351	0.0638
$\mathbf{w}^2$	0.0251	0.8302	0.0000	0.1323	0.0123
$\mathbf{w}^3$	0.0000	0.4078	0.2735	0.0000	0.3187
$\mathbf{w}$	0.0000	0.8068	0.0000	0.1226	0.0706

5.4. RUL prediction results

5.4.1. Initial CV for weight optimization

After obtaining the degradation-dependent weight vectors, the ensemble prediction result of each stage can be calculated using the weighted-sum formulation in Eq. (3). Table 8 compares the RUL prediction errors on the CV units by the original ensemble (EDI) and the proposed ensemble (EDD). As can be seen in the table, the proposed method (EDD) produced smaller CV errors than the original method (EDI) at all the degradation stages, and the error reductions were respectively 1.81%, 0.77% and 17.81% for stages 1, 2 and 3. This resulted in a 2.88% reduction in the overall CV error that was computed with all the 547 CV-testing units. It was expected for EDD to achieve a reduction in the CV error at each stage, since the optimization of degradation-dependent weights (EDD) was performed separately for different stages while that of degradation-independent weights (EDI) was performed once for all stages (i.e., EDD could tune a larger number of weights for minimizing the CV errors than EDI).

5.4.2. Repeated CV for performance validation

In order to validate the performance improvement by the proposed ensemble prognostics method, we independently generated another set of synthetic CV-testing units (i.e., the CV-validation data set) by randomly truncating the VHI curves of the 218 training units. The 10 CV trials were repeated with the CV-validation data set, and the weight vectors ($\mathbf{w}^{1\text{-}3}$ and $\mathbf{w}$) obtained from the initial CV (see Table 7) were used to compute the CV errors by EDI and EDD. This repeated CV allowed the performance of EDI and EDD to be evaluated with a data set that was not used to optimize the weights for these two methods. Tables 5 and 6 summarize the CV errors on the CV-validation data set. In Table 9, the CV-validation errors of the five member algorithms are provided. As can be seen in Table 5, the CV error of RNN is the largest in stage 1 and in Stages 2 and 3 the worst is QB. The SBI-based algorithms (i.e., RS, ES and SS) provide accurate and stable prediction performance on the CV-validation data set. It can be noticed that RNN performs good in stage 3. These observations indicate the algorithm diversity.

In Table 10, it can be seen that the CV-validation errors by the original (EDI) and proposed (EDD) ensemble prognostics methods are similar to the CV-testing errors in Table 8. The proposed ensemble (EDD) outperforms the original ensemble (EDI) with respect to the RUL prediction accuracy. It should be noted that the weight vectors ($\mathbf{w}^{1\text{-}3}$ and $\mathbf{w}$) obtained from the initial CV were used in the CV-validation. Therefore, the

Table 8
CV-testing errors by the original (EDI) and proposed (EDD) ensemble prognostics methods.

Degradation stage	Range of current VHI	No. of CV-testing units	CV-testing error (score function)	
			EDI	EDD
1	[0.7, 1.2]	127	19.8633	19.5041
2	[0.4, 0.7]	202	6.5542	6.5038
3	[−0.2, 0.4]	218	1.6841	1.3841
Overall	[−0.2, 1.2]	547	7.7033	7.4814

Table 9
CV-validation errors by member algorithms for each degradation stage.

Degradation stage	Range of current VHI	No. of CV-validation units	CV-validation error (score function)				
			RS	ES	SS	QB	RNN
1	[0.7, 1.2]	127	26.3345	23.7835	22.7590	45.2030	101.1577
2	[0.4, 0.7]	202	7.3480	5.7521	6.8246	408.6608	60.5555
3	[−0.2, 0.4]	218	2.0100	1.6208	1.6855	114.2616	3.8403

Table 10
CV-validation errors by the original (EDI) and proposed (EDD) ensemble prognostics methods.

Degradation stage	Range of current VHI	No. of CV-validation units	CV- validation error (score function)	
			EDI	EDD
1	[0.7, 1.2]	127	21.4760	20.8857
2	[0.4, 0.7]	202	5.2844	5.1174
3	[−0.2, 0.4]	218	1.8780	1.6033
Overall	[−0.2, 1.2]	547	7.6817	7.3738

Table 11
Procedure of the validation process.

Step 1	Classify the degradation stage of each testing units into the three stages based on the filtered VHI curve. The stage of each unit is determined by the cycle number of the end point of the filtered VHI curve.
Step 2	Estimate the RULs of the units in each stage using the five member algorithms. The prediction matrixes $\hat{\mathbf{L}}^s$ ($s = 1, 2,$ and 3), correspond to the three degradation stages, respectively.
Step 3	Perform the weighted-sum formulation (Eq. (3)). By multiplying the weight vector $\mathbf{w}^s$ in Table 7 with $\hat{\mathbf{L}}^s$ ($s = 1, 2,$ and 3), the ensemble prediction for each degradation stage can be obtained.
Step 4	Calculate validation errors of the proposed ensemble method (EDD).
Step 5	Compare the proposed ensemble (EDD) with the original ensemble (EDI). The weight vector $\mathbf{w}$ in Table 7 is used for the EDI ensemble.
Step 6	Compare the ensemble prediction results of the proposed ensemble (EDD) with those in Ref. [12]. The weight vector $\mathbf{w}_r$ obtained in [12] is used for the comparison, i.e., $\mathbf{w}_r = [0.0000, 0.0470, 0.7462, 0.2068, 0.0000]^T$.

Table 12
Validation errors by the original (EDI) and proposed (EDD) ensemble prognostics methods.

Degradation stage	Range of current VHI	No. of testing units	Validation error (score function)		
			EDI	Ref. [12]	EDD
1	[0.7, 1.2]	85	10.6208	8.6481	**8.0142**
2	[0.4, 0.7]	87	5.7869	**5.7593**	6.0105
3	[−0.2, 0.4]	46	1.0845	1.1790	**1.0703**
Overall	[−0.2, 1.2]	218	6.6794	6.1955	**5.7493**

CV-validation results demonstrate that the proposed ensemble (EDD) is effective and robust for RUL prediction improvement.

5.4.3. Testing data set for performance validation

In order to highlight the effectiveness of the proposed ensemble (EDD) method, the 218 testing data sets of the 2008 IEEE PHM Data Challenge were used to examine the EDD method. The time series signals of the testing data sets were pruned some time prior to a system failure [12]. The objective of validation aims to evaluate the robustness and precision of ensemble prediction improvement by utilizing the degradation-dependent weights $\mathbf{w}^{1\text{-}3}$ (shown in Table 7) on the testing data sets. The validation process is introduced in Table 11.

Table 12 depicts the validation results on the testing data sets using ensemble prediction methods and Fig. 9 shows the prediction plot. In the validation, there were 85, 87, and 46 units in stages 1–3, respectively. As can be seen in Table 12, the validation errors of the proposed ensemble (EDD) in stages 1, stage 2 and overall are smaller than that of the original ensemble (EDI). Although in stage 2 the best prediction result is generated by [12], the overall prediction precision of the EDD ensemble is higher than that of the EDI ensemble. The reason why the proposed ensemble (EDD) is not better/as good to its competitors in stage 2 is probably because of lacking sufficient testing units. A small number of the testing unit in this stage may limit the error distribution and diversity of the validation. Because in the CV-validation in Section 4.4.2 the validation units in each stage is much more than the testing units in this validation process and the proposed ensemble (EDD) achieved better prediction performance in all stages in the CV-validation, it is reasonable to believe that if the testing units in stage 2 increase the EDD ensemble will provide better prediction accuracy than the EDI ensemble. Moreover, it is encouraging to see that the EDD en-

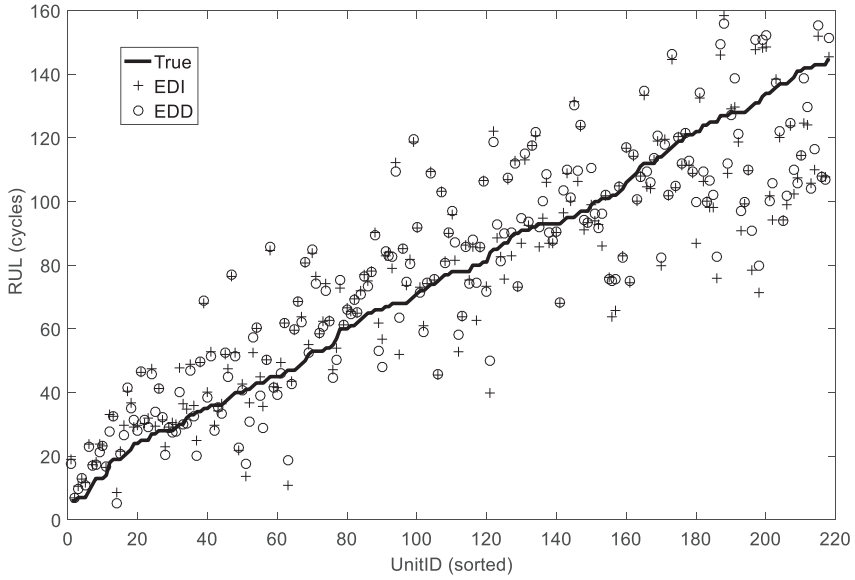

Fig. 9. RUL predictions of 218 testing units for 2008 IEEE PHM Challenge Competition.

semble presents better RUL estimation than its competitors in stage 3 even the number of testing units was only 46. This observation suggests strong robustness ability of the proposed ensemble (EDD). Based on the above CV-validation and testing validation results, it can be safe to conclude that the proposed ensemble method (EDD) is effective and robust for RUL prediction improvement in this case study.

6. Conclusions and future work

In this paper, an ensemble learning-based prognostic approach with degradation-dependent weights was introduced to account for the effects of time-dependent degradation on prognostic accuracy. This new ensemble prognostics method classifies the degradation stages of an entire degradation process using locally weighted linear regression, then determines the optimal degradation-dependent weights by minimizing cross-validation training errors (only during offline training), and finally assigns the degradation-dependent weights to the member prognostic algorithms of an ensemble. To demonstrate the effectiveness of this new method, two case studies were conducted to predict the RULs of an aircraft bearing (Case 1) and an aircraft engine (Case 2). The experimental results have shown that this new method is capable of outperforming the original ensemble prognostics method by accounting for the effects of time-dependent degradation on prognostic accuracy. As mentioned in Section 3.3.1, the focus of this paper was to accurately predict the RUL of an engineered system by accounting for the effects of different degradation stages over the entire system lifecycle instead of varying operation conditions. Two specific objectives were (1) to classify (offline) and identify (online) these degradation stages and (2) to identify (offline) and deploy (online) optimal weights for the member algorithms of an ensemble at each of these stages. In the future, an adaptive weight optimization algorithm will be developed to partition the entire degradation process into multiple degradation stages more accurately.

Acknowledgments

This research was in part supported by the US National Science Foundation (NSF) grant nos. CNS-1566579 and ECCS-1611333, and the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (USDOT/OST-R) through the Midwest Transportation Center (MTC). Any opinions, findings or conclusions in this paper are those of the authors and do not necessarily reflect the views of the sponsoring agency.

References

[1] ISO 13381-1. Condition monitoring and diagnostics of machines—prognostics—part 1: general guidelines. International Standards Organization International Standards Organization Std.; 2004.
[2] Lu L, Han X, Li J, Hua J, Ouyang M. A review on the key issues for lithium-ion battery management in electric vehicles. J Power Sources 2013;226:272–88.
[3] Pecht M. Prognostics and health management of electronics. Hoboken, NJ, USA: John Wiley & Sons; 2008.
[4] Liao L, Köttig F. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. IEEE Trans Reliab 2014;63(1):191–207.
[5] Orchard M, Vachtsevanos G. A particle filtering approach for on-line failure prognosis in a planetary carrier plate. Int J Fuzzy Logic Intell Syst 2007;7(4):221–7.
[6] Saha B, Goebel K, Poll S, Christophersen J. Prognostics methods for battery health monitoring using a Bayesian framework. IEEE Trans Instrum Meas 2009;58(2):291–6.
[7] Swanson DC, Spencer JM, Arzoumanian SH. Prognostic modelling of crack growth in a tensioned steel band. Mech Syst Signal Process 2000;14(5):789–803.
[8] Medjaher K, Tobon-Mejia D, Zerhouni N. Remaining useful life estimation of critical components with application to bearings. IEEE Trans Reliab 2012;61(2):292–302.
[9] Wang T, Yu J, Siegel D, Lee J. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. International conference on prognostics and health management, Denver, CO, October 6–9; 2008.
[10] Coble JB, Hines JW. Prognostic algorithm categorization with PHM challenge application. IEEE, International Conference on Prognostics and Health Management, Denver, CO, October 6–9; 2008.
[11] Heimes FO. Recurrent neural networks for remaining useful life estimation. IEEE, International Conference on Prognostics and Health Management, Denver, CO, October 6–9; 2008.
[12] Hu C, Youn B, Wang P, Yoon J. Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. Reliab Eng Syst Saf 2012;103:120–35.
[13] Li Z, Wu D, Hu C, Terpenny J, Shen S. An ensemble learning-based prognostic approach with degradation-dependent weights: prediction of remaining useful life for aircraft engines. In: Proceedings of the ASME 2017 international design engineering technical conferences & computers and information in engineering conference IDETC/CIE 2017, August 6–9, 2017, Cleveland, Ohio, USA; 2017.
[14] Perrone MP, Cooper LN. When networks disagree: ensemble methods for hybrid neural networks 1992 doi: 10.1.1.32.3857.
[15] Zhang C, Lim P, Qin A, Tan K. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. IEEE Trans Neural Netw Learn Syst 2016;41(1):2306–18.
[16] Sun J, Zuo H, Yang H, Pecht M. Study of ensemble learning-based fusion prognostics. In: Proceedings of 2010 IEEE prognostics and health management conference, PHM2010; 2010.
[17] Xing Y, Ma E, Tsui K, Pecht M. An ensemble model for predicting the remaining useful performance of lithium-ion batteries. Microelectron Reliab 2013;53(6):811–20.
[18] Zhang B, Zhang L, Xu J. Remaining useful life prediction for rolling element bearing based on ensemble learning. Chem. Eng. Trans. 2013;33:157–62.
[19] Baraldi P, Zio E, Mangili F, Gola G, Nystad B. Ensemble of Kernel regression models for assessing the health state of choke valves in offshore oil platforms. Int J Comput Intell Syst 2014;7(2):225–41.
[20] Baraldi P, Mangili F, Zio E. A belief function theory based approach to combining different representation of uncertainty in prognostics. Inf Sci 2015;303:134–49.

[21] Wu D, Jennings C, Terpenny J, Kumara S. Cloud-based machine learning for predictive analytics: prediction of tool wear. In: Proceedings of 2016 IEEE international conference on big data; 2016.
[22] Peel L. Data driven prognostics using a Kalman filter ensemble of neural network models. In: Proceedings of 2008 international conference on prognostics and health management, PHM 2008; 2008.
[23] Peng Y, Dong M. A hybrid approach of HMM and grey model for age-dependent health prediction of engineering assets. Expert Syst Appl 2011;38(10):12946–53.
[24] Liu D, Luo Y, Peng Y, Peng X, Pecht M. Lithium-ion battery remaining useful life estimation based on nonlinear AR model combined with degradation feature. In: Proceedings of 2012 international conference on prognostics and health management, PHM 2012; 2012.
[25] Lim P, Goh C, Tan K, Dutta P. "Multimodal degradation prognostics based on switching kalman filter ensemble. IEEE Trans Neural Netw Learn Syst 2017;28(1):136–48.
[26] Liu J, Vitelli V, Zio E, Seraoui R. A novel dynamic-weighted probabilistic support vector regression-based ensemble for prognostics of time series data. IEEE Trans Reliab 2015;64(4):1203–13.
[27] Chen X, Yu J, Tang D, Wang Y. Remaining useful life prognostic estimation for aircraft subsystems or components: a review. In: Proceedings of IEEE 10th international conference on electronic measurement and instruments, ICEMI 2011; 2011. p. 94–8.
[28] Alam M, Bodruzzaman M, Zein-Sabatto M. Online prognostics of aircraft turbine engine component's remaining useful life (RUL). In: Proceedings of IEEE SOUTHEASTCON; 2014.
[29] Liu J, Zhang M, Zuo H, Xie J. Remaining useful life prognostics for aeroengine based on superstatistics and information fusion. Chin J Aeronaut 2014;27(5):1086–96.
[30] Liu J, Xie J, Zuo H, Zhang M. Residual lifetime prediction for aeroengines based on Wiener process with random effects. Acta Aeronautica et Astronautica Sinica 2015;36(2):564–74.
[31] Lasheras F, Nieto P, de Cos Juez F, Bayón R, Suárez V. A hybrid PCA-CART–MARS-based prognostic approach of the remaining useful life for aircraft engines. Sensors 2015;15(3):7062–83.
[32] Lim P, Goh C, Tan K. A time window neural network based framework for remaining useful life estimation. In: Proceedings of international joint conference on neural networks, IJCNN 2016; 2016. p. 1746–53.
[33] Bluvband Z, Porotsky S. RUL prognostics and critical zone recognition for suspended time-series. In: Proceedings of IEEE conference on prognostics and health management, PHM 2015; 2015.
[34] Yuan M, Wu Y, Lin L. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. In: Proceedings of IEEE/CSAA international conference on aircraft utility systems; 2016. p. 135–40.
[35] Yan H, Liu K, Zhang X, Shi J. "Multiple sensor data fusion for degradation modeling and prognostics under multiple operational conditions. IEEE Trans Reliab 2016;65(3):1416–26.
[36] Wang P, Gao R. "Markov nonlinear system estimation for engine performance tracking. ASME J Eng Gas Turbines Power 2016;138(9):091201.
[37] Vijayakumar S, Aaron D, Schaal S. Incremental online learning in high dimensions. Neural Comput 2005;7(12):2602–34.
[38] Wang T, Yu J, Siegel D, Lee J. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In: Proceedings of international conference on prognostics and health management, PHM 2008; 2008.
[39] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. In: International conference on prognostics and health management; 2008. p. 1–9.
[40] Tipping ME. Sparse Bayesian learning and the relevance vector machine. J Mach Learn Res 2001;1:211–44.
[41] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput 2004;14(3):199–222.
[42] Wang M, Vandermaar AJ, Srivastava KD. Review of condition assessment of power transformers in service. IEEE Electr Insul Mag 2002;18(6):12–25.
[43] Cernansky M, Makula M, Cernansky L. Organization of the state space of a simple recurrent network before and after training on recursive linguistic structures. Neural Netw 2007;20(2):236–44.
[44] Qian Y, Yan R, Gao R. A multi-time scale approach to remaining useful life prediction in rolling bearing. Mech Syst Sig Process 2017;83:549–67.
[45] Chelidze D, Cusumano JP. A dynamical systems approach to failure prognosis. J Vib Acoust 2004;126(1):2–8.
[46] Del Moral P. Non-linear filtering: interacting particle resolution. Markov Processes Relat Fields 1996;2(4):555–81.
[47] Xi Z, Jing R, Wang P, Hu C. A copula-based sampling method for data-driven prognostics. Reliab Eng Syst Saf 2014;132:p72–82.
[48] Baraldi P, Compare M, Sauco S, Zio E. Ensemble neural network-based particle filtering for prognostics. Mech Syst Sig Process 2013;41(1):288–300.
[49] Sbarufatti C, Corbetta M, Manes A, Giglio M. Sequential Monte-Carlo sampling based on a committee of artificial neural networks for posterior state estimation and residual lifetime prediction. Int J Fatigue 2016;83:10–23.
[50] Cadini F, Sbarufatti C, Corbetta M, Giglio M. A particle filter-based model selection algorithm for fatigue damage identification on aeronautical structures. Struct Control Health Monit 2017;24(11). doi:10.1002/stc.2002.