Reliability Engineering and System Safety 103 (2012) 120–135

Contents lists available at SciVerse ScienceDirect

Reliability Engineering and System Safety

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

Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life

Chao Hu [a], Byeng D. Youn [b,*], Pingfeng Wang [c], Joung Taek Yoon [b]

[a] Department of Mechanical Engineering, the University of Maryland at College Park, College Park, MD 20742, USA (Currently at Medtronic Inc.)
[b] School of Mechanical and Aerospace Engineering, the Seoul National University, Seoul 151–742, South Korea
[c] Department of Industrial and Manufacturing Engineering, the Wichita State University, Wichita, KS 67260, USA

ARTICLE INFO

Article history:
Received 29 March 2011
Received in revised form
13 February 2012
Accepted 8 March 2012
Available online 17 March 2012

Keywords:
Ensemble
k-fold cross validation
Weighting schemes
Data-driven prognostics
RUL prediction

ABSTRACT

Prognostics aims at determining whether a failure of an engineered system (e.g., a nuclear power plant) is impending and estimating the remaining useful life (RUL) before the failure occurs. The traditional data-driven prognostic approach is to construct multiple candidate algorithms using a training data set, evaluate their respective performance using a testing data set, and select the one with the best performance while discarding all the others. This approach has three shortcomings: (i) the selected standalone algorithm may not be robust; (ii) it wastes the resources for constructing the algorithms that are discarded; (iii) it requires the testing data in addition to the training data. To overcome these drawbacks, this paper proposes an ensemble data-driven prognostic approach which combines multiple member algorithms with a weighted-sum formulation. Three weighting schemes, namely the accuracy-based weighting, diversity-based weighting and optimization-based weighting, are proposed to determine the weights of member algorithms. The *k*-fold cross validation (CV) is employed to estimate the prediction error required by the weighting schemes. The results obtained from three case studies suggest that the ensemble approach with any weighting scheme gives more accurate RUL predictions compared to any sole algorithm when member algorithms producing diverse RUL predictions have comparable prediction accuracy and that the optimization-based weighting scheme gives the best overall performance among the three weighting schemes.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

To support critical decision-making processes such as maintenance replacement and system design, activities of health monitoring and life prediction are of great importance to high-risk engineered systems composed of multiple components, complex joints, and various materials, such as aerospace systems, nuclear power plants, chemical plants, advanced military systems and so on. Stressful conditions (e.g., high pressure, high temperature and high irradiation field) imposed on these systems are the direct causes of damage in their integrity and functionality, which necessitates the continuous monitoring of these systems due to the health and safety implications [1–3]. Research on real-time diagnosis and prognosis which interprets data acquired by distributed sensor networks, and utilizes these data streams in making critical decisions provides significant advancements across a wide range of applications. Maintenance and life-cycle management of these high-risk engineered systems for minimizing the cost [4–6], maximizing the availability [7] and extending the service life [8] is one of the beneficiary application areas because of the pervasive nature of monitoring and maintenance activities throughout the manufacturing and service sectors and, especially, the extremely high failure costs. For instance, in nuclear power plants, unexpected breakdowns can be prohibitively expensive and disastrous since they immediately result in lost power production, correct maintenance cost, reduced public confidence, and, possibly, human injuries and deaths. In order to reduce and possibly eliminate such problems, it is necessary to accurately assess current system health condition and precisely predict the remaining useful lives (RULs) of operating components, subsystems, and systems in the high-risk engineered systems.

In general, prognostic approaches can be categorized into model-based approaches [9–13], data-driven approaches [14–18] and hybrid approaches [19–21]. The application of general model-based prognostic approaches relies on the understanding of system physics-of-failure and underlying system degradation models. Myotyri et al. [9] proposed the use of a stochastic filtering technique for real-time RUL prediction in the case of fatigue crack growth while

* Corresponding author.
E-mail addresses: huchaost@umd.edu (C. Hu), bdyoun@snu.ac.kr (B.D. Youn), pingfeng.wang@wichita.edu (P. Wang), kaekol@snu.ac.kr (J. Taek Yoon).

0951-8320/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ress.2012.03.008

considering the uncertainties in both degradation processes and condition monitoring measures. A similar particle filtering approach was later applied to condition-based component replacement in the context of fatigue crack growth [10]. Luo et al. [11] developed a model-based prognostic technique that relies on an accurate simulation model for system degradation prediction and applied this technique to a vehicle suspension system. Gebraeel presented a degradation modeling framework for RUL predictions of rolling element bearings under time-varying operational conditions [12] or in the absence of prior degradation information [13]. As high-risk engineered systems generally consist of multiple components with multiple failure modes, understanding all potential physics-of-failures and their interactions for a complex system is almost impossible. With the advance of modern sensor systems as well as data storage and processing technologies, the data-driven approaches for system health prognostics, which are mainly based on the massive sensory data with less requirement of knowing inherent system failure mechanisms, have been widely used and become popular. A good review of data-driven prognostic approaches was given in [14]. Data-driven prognostic approaches generally require the sensory data fusion and feature extraction, statistical pattern recognition, and for the life prediction, the interpolation [15,16], extrapolation [17], or machine learning [18] and so on. Hybrid approaches attempt to take advantage of the strength from data-driven approaches as well as model-based approaches by fusing the information from both approaches. Kozlowski et al. [19] described a data fusion approach where domain knowledge and predictor performance are used to determine weights for different state-of-charge predictors. Goebel et al. [20] employed a Dempster–Shafer regression to fuse a physics-based model and an experience-based model for prognostics. Saha et al. [21] combined the offline relevance vector machine (RVM) with the online particle filter for battery prognostics. Similar to model-based approaches, the application of hybrid approaches is limited to the cases where sufficient knowledge on system physics-of-failures is available.

Implicit relationship between the RUL and the sensory signals makes it difficult to know which prognostic algorithm performs best in a specific application. Furthermore, there are many factors that affect the prediction accuracy and robustness, such as (i) dependency of the algorithm's accuracy on the number of units in a training data set, (ii) significant variability in manufacturing conditions and large uncertainties in environmental and operational conditions, (iii) the amount of effective sensory signals for RUL predictions, and (iv) the form of degradation trend (e.g., linear, nonlinear, noisy, smooth). Therefore, no single prognostic algorithm works well for all possible situations. Instead of using an individual prognostic algorithm, it would be beneficial to combine multiple algorithms to form a hybrid algorithm.

Combining approximate algorithms into an ensemble, i.e., ensemble methods, was motivated to improve the robustness and accuracy of algorithm in the machine learning community. The methods can be classified by the combining strategy; by consensus or by learning. Examples of noted ensemble methods and brief descriptions are arranged below in Tables 1 and 2 [22].

The ensemble method founds its applications in a wide variety of research fields, such as the development of committees of neural networks [28,29], the metamodeling for the design of modern engineered systems [30–32], the discovery of regulatory motifs in bioinformatics [33], the detection of traffic incidents [34], the transient identification of nuclear power plant [35], and the development of ensemble Kalman filters [36]. However, the utilization of the ensemble approach for the data-driven prognostics is still in infancy. The only relevant work we are aware of comes from [37] where only two data-driven algorithms are employed as member algorithms and their weights are empirically determined based on a one-time training error without a systematic scheme for error estimation and performance validation. Most data-driven prognostic practices select a single algorithm with the best accuracy from the algorithm pool while discarding the others. This approach not only wastes the resource devoted to developing different algorithms, but also suffers from the lack of robustness.

Estimating the accuracy of a prognostic algorithm is important not only for evaluating its prediction accuracy but also for choosing the best algorithm from a given set (algorithm selection), or combining algorithms. Many data-driven approaches [14,16] use the so-called holdout method, which divides the original run-to-failure data set into two mutually exclusive subsets called a training set and a testing set, or holdout set. The holdout method is straightforward and computationally efficient. However, it often produces a large variance of the resulting estimate and requires the testing data set which increases the overall expenses.

To overcome the above shortcomings, this study proposes an ensemble approach that employs the k-fold cross validation (CV) to estimate the accuracy of a given ensemble and proposes three weighting schemes to determine the weight values. Assumptions for this study are listed below:

(1) Sensory data from multiple run-to-failure units are available, either from the computer simulation or field testing.
(2) A single failure mode is considered, i.e., the RUL prediction is exclusively for this failure mode.
(3) The underlying physics of the system fault propagation is not comprehensive or it is too expensive to derive a reliable physical damage model for a complex engineered system. Both cases entail the use of the data-driven prognostics.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to the data-driven prognostic algorithms selected in this study. Section 3 presents the proposed ensemble approach with the k-fold CV and three weighting schemes.

Table 1
Examples of noted ensemble methods.

Combining strategy	Ensemble method	Description	Reference
By consensus	Bagging	Bagging determines a class label with major voting by multiple classifiers.	Breiman (1996) [23]
	Random forest	Random forest improves the performance of bagging by combining with random feature selection scheme.	Breiman (2001) [24]
By learning	Boosting	Boosting trains weak classifiers and combines them into a strong classifier.	Schapire (1990) [25]
	Adaboost	Adaboost trains each base classifier with a weighted data set of which weighting coefficients are computed from classification errors by the previous classifiers, and then aggregates the base classifiers into one.	Freund and Schapire (1997) [26]
	Rule ensemble	Not only use a basis function as a base classifier, a rule ensemble includes a rule as a base classifier. As the rule has a simple form, it is easy to understand the influences of rules on predictions and the degree of dependency on each other.	Friedman and Popescu (2008) [27]

Applications of the proposed methodology are presented in Section 4 and the conclusion of this work is given in Section 5.

2. Description of prognostic algorithms

2.1. A general description of member algorithms

This section provides a brief overview of the five selected data-driven prognostic algorithms: Method 1 – a similarity-based interpolation (SBI) approach with the relevance vector machine (RVM) as the regression technique (RVM–SBI) [15,49], Method 2 – SBI with the support vector machine (SVM) as the regression technique (SVM–SBI) [15,47], Method 3 – SBI with the least-square exponential fitting (Exp–SBI) [15], Method 4 – A Bayesian linear regression with the least-square quadratic fitting (Quad–BLR) [17], and Method 5 – A recurrent neural network (RNN) approach [18,50]. A data processing scheme with a generic health index system is used for the first four algorithms while a data processing scheme with a simple normalization scheme for the last algorithm.

2.2. Methods 1–3: Similarity-based interpolation approaches

2.2.1. Data processing with a generic health index system

Successful implementations of prognostic algorithms require the extraction of the health condition signatures and background health knowledge from massive training/testing sensory signals from engineered system units. To do so, this study will use a generic health index system that is composed of two distinguished health indices: physics health index (PHI) and virtual health index (VHI). In general, the PHI uses a dominant physical signal as a direct health metric and is thus applicable only if sensory signals are directly related to physics-of-failures. In the literature, most engineering practices of health prognostics are based on various PHIs, such as the battery impedance [21], the magnitude of the vibration signal [45] and the radio frequency (RF) impedance [46]. In contrast, the virtual health index (VHI) is applicable even if sensory signals are not directly related to system physics-of-failures. In this study, the VHI system will be employed which transforms the multi-dimensional sensory signals to one-dimensional VHI with a linear data transformation method [15]. The VHI system will be detailed in what follows.

Suppose there are two multi-dimensional sensory data sets that represent the system failed and healthy states, $\mathbf{Q}_0$ of $M_0 \times D$ matrix and $\mathbf{Q}_1$ of $M_1 \times D$ matrix, respectively, where M_0 and M_1 are the data sizes for system failed and healthy states, respectively, and D is the dimension of each dataset. With these two data matrices, a transformation matrix $\mathbf{T}$ can be obtained to transform the multi-dimensional sensory signal into the one-dimensional VHI as

$$\mathbf{T} = (\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T\mathbf{S}_{\text{off}} \tag{1}$$

where $\mathbf{Q}=[\mathbf{Q}_0; \mathbf{Q}_1]$, $\mathbf{S}_{\text{off}}=[\mathbf{S}_0, \mathbf{S}_1]^T$, $\mathbf{S}_0$ is a $1 \times M_0$ zero vector and $\mathbf{S}_1$ is a $1 \times M_1$ unity vector. This transformation matrix $\mathbf{T}$ can transform any sensory signal from the offline learning or online prediction process to the normalized VHI as $\mathbf{H}=\mathbf{Q}_{\text{off}} \cdot \mathbf{T}$ or $\mathbf{H}=\mathbf{Q}_{\text{on}} \cdot \mathbf{T}$, where $\mathbf{Q}_{\text{off}}$ and $\mathbf{Q}_{\text{on}}$ are the offline and online multi-dimensional sensory data sets, respectively, and, if we assume the data sizes for $\mathbf{Q}_{\text{off}}$ and $\mathbf{Q}_{\text{on}}$ are respectively M_{on} and M_{off} (i.e., $\mathbf{Q}_{\text{off}}$ of $M_{\text{off}} \times D$ matrix and $\mathbf{Q}_{\text{on}}$ of $M_{\text{on}} \times D$ matrix), $\mathbf{H}$ will be a column vector of the size M_{off} or M_{on}. The VHI can also be denoted as $h(t_i)$ for $i=1, \ldots, M_{\text{off}}$ (for the offline case) or for $i=1, \ldots, M_{\text{on}}$ (for the online case), varying approximately between 0 and 1. This VHI can be used to construct background health knowledge (e.g., predictive health degradation curves) in the offline training process and to further conduct the online prediction process.

2.2.2. Training with RVM, SVM, or exponential fitting

Three regression techniques, namely, relevance vector machine (RVM), support vector machine (SVM), and least-square exponential fitting, are employed to construct the background health knowledge using the VHI of offline system units obtained from Section 2.2.1. These regression techniques can generate different sets of predictive health degradation curves for offline system units.

2.2.2.1. Relevance vector machine. Suppose we have a training data set $\{t_i, h_i\}$, $i=1, \ldots, M_s$, sampled from a scalar-valued function with additive zero mean Gaussian noise ε with a variance σ^2. The RVM is a special case of a sparse linear model to approximate this data set as

$$h(t) = \sum_{i=1}^{M_s} \omega_i \phi(t,t_i) + \omega_0 \tag{2}$$

where $\boldsymbol{\omega}=(\omega_0, \ldots, \omega_{Ms})^T$ is a kernel weight vector and $\phi(t, t_i)$ is a kernel function centered at the training point t_i. Assuming the independence of $\{h_i\}$, $i=1, \ldots, M_s$, we have the likelihood of the observed data as

$$p(\mathbf{h}|\boldsymbol{\omega},\sigma^2) = (2\pi\sigma^2)^{-M_s/2}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{h}-\boldsymbol{\Phi}\boldsymbol{\omega}\|^2\right) \tag{3}$$

where $h=(h_1, \ldots, h_{Ms})^T$, $\boldsymbol{\Phi}$ is an $M_s \times (M_s+1)$ design matrix constructed the training points $\mathbf{t}$ with $\boldsymbol{\Phi}_{ij}=\phi(t_i, t_j)$.

To evaluate the unknown parameters in Eq. (3) from a Bayesian perspective, a sparse weight prior distribution can be assigned, in such a way that a different variance parameter is assigned for each weight, as

$$p(\boldsymbol{\omega}|\boldsymbol{\alpha}) = \prod_{i=0}^{M_s} N(\omega_i|0,\ \alpha_i^{-1}) \tag{4}$$

where $\boldsymbol{\alpha}=(\alpha_0, \ldots, \alpha_{Ms})$ is a vector consisting of M_s+1 hyper-parameters, which are treated as independent random variables. To specify this hierarchical Bayesian inference model, prior distributions for $\boldsymbol{\alpha}$ and the noise variance σ^2 must be defined. For scale parameters $\boldsymbol{\alpha}$ and σ^2, it is common to use Gamma prior distributions as

$$\begin{aligned} p(\boldsymbol{\alpha}) &= \prod_{i=0}^{M_s} \text{Gamma}(\alpha_i|a,\ b) \\ p(\beta) &= \text{Gamma}(\beta|c,\ d) \end{aligned} \tag{5}$$

where $\beta=\sigma^{-2}$, $\text{Gamma}(\alpha|a, b)=\Gamma(a)^{-1}b^a\alpha^{a-1}e^{-b\alpha}$ with $\Gamma(a)$ being the gamma function, a, b, c and d are the hyper-parameters and set to small values to form a flat Gamma prior.

With the prior defined, the posterior distribution over the weights is given by the Bayesian inference as

$$p(\boldsymbol{\omega}|\mathbf{h},\boldsymbol{\alpha},\sigma^2) = (2\pi)^{-(M_s+1)/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\left(-\tfrac{1}{2}(\boldsymbol{\omega}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\omega}-\boldsymbol{\mu})\right) \tag{6}$$

where the posterior mean vector of the weights is $\boldsymbol{\mu}=\sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}\mathbf{h}$, and the covariance matrix is $\boldsymbol{\Sigma}=(\sigma^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}+\mathbf{A})^{-1}$ with $\mathbf{A}=\text{diag}(\alpha_0, \ldots, \alpha_{Ms})$. Appropriate iterative optimization methods [49], such as marginal likelihood optimization, expectation maximization (EM) algorithms or incremental optimization algorithms, can be employed to find the hyper-parameter posterior modes or most probable values $\boldsymbol{\alpha}_m$ and σ_m^2 that maximize $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{h}) \propto p(\mathbf{h}|\boldsymbol{\alpha}, \sigma^2)\ p(\boldsymbol{\alpha})\ p(\sigma^2)$. More detailed descriptions of the RVM can be found in [49].

2.2.2.2. Support vector machine. Similar to the RVM, the SVM is also a special case of a sparse linear model that approximates the data set $\{t_i, h_i\}$, $i=1, \ldots, M_s$, as

$$h(t) = \sum_{i=1}^{M_s} \omega_i \phi(t,t_i) + \omega_0 \tag{7}$$

where the linear or nonlinear kernel function $\phi(t, t_i)$ are centered at the training point t_i. The optimum regression function can be obtained by solving the following optimization problem

$$\begin{aligned}&\text{Minimize}\ \ \frac{1}{2}|w|^2+C\sum_{i=1}^{M_s}(\xi_i^-+\xi_i^+)\\&\text{Subject to}\ \ h_i-\sum_{j=1}^{M_s}\omega_i\phi(t_i,t_j)-\omega_0\le\varepsilon+\xi_i^+\\&\sum_{j=1}^{M_s}\omega_i\phi(t_i,t_j)+\omega_0-h_i\le\varepsilon+\xi_i^-\\&\xi_i^+,\xi_i^-\ge 0, i=1,\ldots,M_s\end{aligned}\tag{8}$$

where the regularization parameter C specifies the tradeoff between the flatness and tolerance; ξ_i^- and ξ_i^+ are slack variables defining the upper and lower constraints on the predictions with an ε-insensitive loss function. With the Karush–Kuhn–Tucker (KKT) conditions, the optimization problem can be reformulated as

$$\begin{aligned}&\text{Maximize}\ \ -\tfrac{1}{2}\sum_{i=1}^{M_s}\sum_{j=1}^{M_s}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\phi(t_i,t_j)-\sum_{i=1}^{M_s}(\alpha_i+\alpha_i^*)\cdot\varepsilon+\sum_{i=1}^{M_s}(\alpha_i-\alpha_i^*)y_i\\&\text{Subject to}\ \ \sum_{i=1}^{M_s}(\alpha_i-\alpha_i^*)=0, 0\le\alpha_i,\alpha_i^*\le C,\quad i=1,\ldots,M_s\end{aligned}\tag{9}$$

The weights and bias term can be computed as

$$\begin{aligned}&w_i=\alpha_i-\alpha_i^*,\quad i=1,\ldots,M_s\\&w_0=-\frac{1}{2}\sum_{i=1}^{M_s}(\alpha_i-\alpha_i^*)\left[\phi(x_i,x_r)+\phi(x_i,x_s)\right]\end{aligned}\tag{10}$$

Then, the regression function can be expressed as

$$h(t)=\sum_{i=1}^{M_s}(\alpha_i-\alpha_i^*)\phi(t,t_i)+\omega_0\tag{11}$$

Detailed information regarding the use of SVM for regression can be found in [47]. This study used the MATLAB® program developed by Rakotomamonjy [48].

2.2.2.3. Exponential fitting. Compared to the RVM and SVM, an exponential fitting is much simpler and easier to implement. In this study, an exponential function was constructed as

$$h(t)=b_1[\exp(b_2t)-1]\tag{12}$$

where the optimum coefficients b_1 and b_2 can be obtained that minimize the least-square error of the exponential fitting.

2.2.3. RUL prediction using SBI

The RUL prediction process involves two procedures: (i) determination of an initial health condition and (ii) the RUL prediction using the similarity-based interpolation (SBI). This process closely follows the similarity-based approach in [15].

2.2.3.1. Determination of initial health condition. Different online testing units often exhibit different initial health indices due to different initial health conditions. Thus, accurate estimations of initial health conditions for online testing units are of great importance to precise RUL predictions. Based on the predictive health degradation curve ($\mathbf{h}_p$) from an offline system unit, an optimum fitting is conducted to determine a time-scale initial health condition (T_0) that minimizes the sum of squared differences *SSD* between the online and offline health index data. The optimum fitting can be formulated as

$$\begin{aligned}&\text{Minimize}\ SSD=\sum_{j=1}^{M_s}(h_r(t_j)-h_p(t_j+T_0))^2\\&\text{subject to}\ T_0\in[0, TS-\Delta t]\end{aligned}\tag{13}$$

where $h_r(t_j)$ and $h_p(t_j)$ are the online and predictive health indices at t_j, respectively, M_s is the length of the online health index data; T_0 is the time-scale initial health condition; Δt is the time span ($=t_{Ms}-t_1$) of the online health index data; *TS* is the time span of a predictive health degradation curve, i.e., the life span of an offline system unit. It is noted that the predictive health degradation curve $\mathbf{h}_p$ built in the offline training process (see Section 2.2.2) is essentially the regression model in Eqs. (2), (7) or (12). This optimization process basically moves the online health index data $\mathbf{h}_r$ along the time axis of a predictive health degradation curve $\mathbf{h}_p$ to find the optimum time-scale initial health state (T_0) that best matches $\mathbf{h}_r$ with $\mathbf{h}_p$ with respect to the *SSD*. Assuming that the data size of the offline system unit is M_{off}, it follows that, among M_{off} offline data points on the offline health degradation curve, we only select M_s consecutive data points overlapping with the online data along the time axis to compute the *SSD*. Once T_0 is determined, the projected remaining life of an online system unit based on a given predictive health degradation curve can be calculated as

$$L^P=TS-\Delta t-T_0\tag{14}$$

Repeating the optimum fitting process on K predictive health degradation curves from K different offline system units gives K RUL estimates (L_i^P for $i=1, \ldots, K$).

2.2.3.2. Similarity-based interpolation. In the similarity-based interpolation (SBI), the predictive RUL is a linear interpolation function in terms of different projected RULs (L_i for $i=1, \ldots, K$) of an online unit as

$$L=\frac{1}{W}\sum_{i=1}^{K}(W_i\cdot L_i)\quad\text{where}\ W=\sum_{i=1}^{K}W_i\tag{15}$$

where L_i is the projected RUL on the ith predictive health degradation curve (for simplicity, we use L_i instead of L_i^P here), W_i is the ith similarity weight. A similarity weight W_i can be defined as the inverse of the corresponding SSD_i, i.e., $W_i=(SSD_i)^{-1}$. This definition ensures that a greater similarity gives a greater weight.

2.3. Method 4: Extrapolation-based approach

Unlike the similarity-based interpolation (SBI) approach, the extrapolation-based approach employs the training data set not for the comparison with the testing data set but rather for obtaining prior distributions of the degradation model parameters. The testing data set is then used to update these prior distributions. An RUL estimate can be obtained by extrapolating the updated degradation model to a predefined failure threshold. For the construction and updating of the degradation model, this study employed the Bayesian linear regression method used in [17].

2.3.1. Data processing with a generic health index system

This section follows exactly the descriptions in Section 2.2.1 and will not be detailed here.

2.3.2. Training with quadratic fitting

In this study, the least-square quadratic fitting was employed as the degradation model for both the training and testing units.

A quadratic function was constructed as

$$h(t) = b_1 t^2 + b_2 t + b_3 \tag{16}$$

where the optimum coefficients b_1, b_2 and b_3 can be obtained that minimize the least-square error of the quadratic fitting. The optimum model parameters for a data set $\{t_i, h_i\}$, $i=1, \ldots M_s$, are estimated as

$$\mathbf{b} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{h} \tag{17}$$

where the coefficient vector $\mathbf{b}=[b_1, b_2, b_3]^T$, the design matrix $\mathbf{\Phi}$ is of the size $M_s \times 3$ with $\mathbf{\Phi}_{ij}=t_i^j$, the diagonal covariance matrix $\mathbf{\Sigma}$ is of the size $M_s \times M_s$ with the diagonal elements $\mathbf{\Sigma}_{ij}=\sigma^2$ and the off-diagonal elements being zero, the VHI vector $\mathbf{h}=[h_1, \ldots, h_{Ms}]^T$. For the training process without prior information, the value of σ^2 does not affect the least-square fitting results and we simply set $\sigma^2=1$.

2.3.3. RUL prediction with Bayesian linear regression

The Bayesian linear regression employs the Bayesian updating technique to update the prior distributions of the degradation parameters $\mathbf{b}$ and the RUL prediction is then accomplished by extrapolating the model shown in Eq. (16) with the posterior distributions of $\mathbf{b}$. The Bayesian linear regression method for RUL prediction is briefly explained here and a more complete discussion about this method can be found in references [17,53]. The prior information of $\mathbf{b}$ can be obtained from the training data, i.e., the prior distribution of b_j is represented by a normal distribution, $N(\mu_j, \sigma_j^2)$, with the mean μ_j and the standard deviation σ_j, $j=1, 2, 3$. The value of σ_j^2 can be estimated from the training data set as the root mean square error between predicted and true degradation curves. In order to include this prior information into the regression estimation of model parameters b as shown in Eq. (17), the design matrix $\mathbf{\Phi}$, the VHI vector $\mathbf{h}$, and the diagonal covariance matrix $\mathbf{\Sigma}$ need to be changed accordingly. As there are three parameters in total, b_1, b_2 and b_3, the design matrix $\mathbf{\Phi}$ constructed with a testing data set is appended with three additional rows with $\mathbf{\Phi}_{(Ms+j)j}=1$, $j=1, 2, 3$, and all other elements being zero. Accordingly the VHI vector $\mathbf{h}$ is appended with three additional rows with $\mathbf{h}_{Ms+j}=\mu_j$, $j=1, 2, 3$, and the covariance matrix $\mathbf{\Sigma}$ is augmented with three additional rows and columns with $\mathbf{\Sigma}_{(Ms+j)(Ms+j)}=\sigma_j^2$, $j=1, 2, 3$, and all other elements being zero. With the updated design matrix $\mathbf{\Phi}$, the VHI vector $\mathbf{h}$, and the diagonal covariance matrix $\mathbf{\Sigma}$, the posterior estimate of the degradation parameters bi can be obtained with Eq. (17), and the updated quadratic degradation model shown in Eq. (16) can be extrapolated to the degradation threshold $h_c=0$ to obtain the *RUL* as:

$$RUL = t - t_0 : \{\forall t \in [t_0, +\infty) \,|\, b_1 t^2 + b_2 t + b_3 = 0\} \tag{18}$$

where t_0 is the time that the system has been operating until the RUL prediction is made.

2.4. Method 5: Recurrent neural network approach

2.4.1. Data processing with a normalization scheme

As a standard data processing technique, the normalization provides a common scale among all the dimensions of a data set. Suppose that a multi-dimensional sensory data set $\mathbf{Q}$ in the form of a matrix of the size $M_s \times D$ comes from the same operation condition, where M_s is the data size and D is the dimension. The normalized data set $\mathbf{Q}^N$ of the data set $\mathbf{Q}$ can be expressed as

$$\mathbf{Q}_{ij}^N = \frac{\mathbf{Q}_{ij} - \mu_j}{\sigma_j} \tag{19}$$

where $\mathbf{Q}_{ij}^N$ is the normalized value of $\mathbf{Q}_{ij}$, μ_j and σ_j are the mean and standard deviation of the jth dimension of the data set $\mathbf{Q}$, respectively.

2.4.2. Recurrent neural network

The RNN is capable of learning the nonlinear dynamic temporal behavior due to the use of an internal state and feedback. A first-order simple RNN is an example of multi-layer perceptron (MLP) with feedback connections (see Fig. 1). The network is composed of four layers, namely, the input layer I, recurrent layer R, context layer C and output layer O. Units of the input layer and the recurrent layer are fully connected through the weights $\mathbf{W}^{RI}$ while units of the recurrent layer and output layer are fully connected through the weights $\mathbf{W}^{OR}$. Through the recurrent weights $\mathbf{W}^{RC}$, the time delay connections link current recurrent units $\mathbf{R}^{(t)}$ with the context units $\mathbf{C}^{(t)}$ holding recurrent units $\mathbf{R}^{(t-1)}$ in the previous time step. Let $\mathbf{I}^{(t)}=(I_1^{(t)}, \ldots, I_j^{(t)}, \ldots, I_{|I|}^{(t)})$, $\mathbf{R}^{(t)}=(R_1^{(t)}, \ldots, R_j^{(t)}, \ldots, R_{|R|}^{(t)})$ and $\mathbf{O}^{(t)}=(O_1^{(t)}, \ldots, O_j^{(t)}, \ldots, O_{|O|}^{(t)})$ be the input patterns, recurrent activities and output activities at the time step t, respectively, where $|I|$, $|R|$ and $|O|$ denote the numbers of the input, recurrent and output units, respectively. The net input of the ith recurrent unit can be computed as

$$\tilde{R}_i^{(t)} = \sum_j W_{ij}^{RI} I_j^{(t)} + \sum_j W_{ij}^{RC} R_j^{(t-1)} \tag{20}$$

Given the logistic sigmoid function as the activation function f, the output activity of the ith recurrent unit can then be computed as

$$R_i^{(t)} = f(\tilde{R}_i^{(t)}) = \left[1 + \exp(-\tilde{R}_i^{(t)})\right]^{-1} \tag{21}$$

The net input and output activity of the ith output unit can be computed, respectively, as

$$\tilde{O}_i^{(t)} = \sum_j W_{ij}^{OR} R_j^{(t)} \tag{22}$$

and

$$O_i^{(t)} = f\left(\tilde{O}_i^{(t)}\right) = \left[1 + \exp\left(-\tilde{O}_i^{(t)}\right)\right]^{-1} \tag{23}$$

In this study, the inputs to the RNN are the normalized sensory data set $\mathbf{Q}^N$ and the outputs are the RULs associated with the data set. We used the MATLAB® program developed by Cernansky [50]. In the RNN training process, the back propagation through time and extended Kalman filter were used to calculate the gradients of network weights and to update network weights, respectively.

3. Ensemble of prognostic algorithms

It is essential to propose a robust prognostic solution that accurately predicts the RUL using data features extracted from multi-dimensional sensory degradation signals. For building such a unified structural health prognostic framework, this paper proposes (i) a weighted-sum formulation for an ensemble of prognostic algorithms, (ii) k-fold cross validation (CV) to evaluate the error metric associated with a candidate ensemble model; and (iii) three weighting schemes to determine the weight values for the member algorithms. This section is organized as follows. Section 3.1 presents the basic weighted-sum formulation for the RUL prediction. Section 3.2 describes the background of the k-fold CV and how it can be applied for estimating the accuracy of a prognostic algorithm. Section 3.3 describes the three proposed weighting schemes. The overall procedure of the ensemble approach is described in Section 3.4.

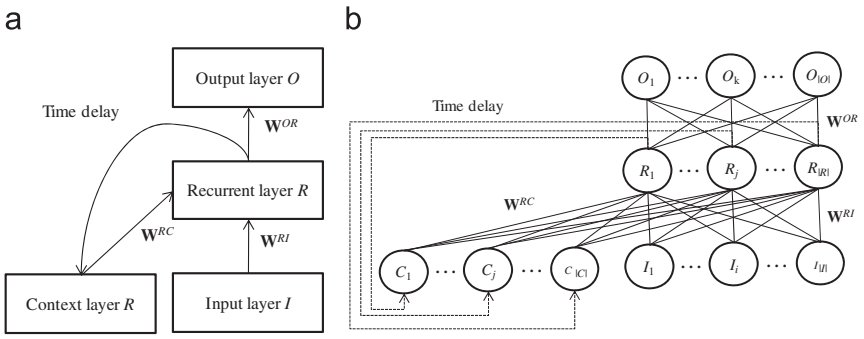

Fig. 1. Simplified (a) and more detailed representation (b) of Elman's simple RNN [50].

3.1. Weighted-sum formulation

A simple average of RUL predictions obtained using the member algorithms means assigning equal weights to the member algorithms used for prognostics. This is acceptable only when the member algorithms provide the same level of accuracy for a given problem. However, it is more likely that an algorithm tends to be more accurate than others. It is ideal to assign a greater weight to a member algorithm with higher prediction accuracy in order to enhance its prediction accuracy and robustness. Hence, member algorithms with different prediction performance should be multiplied by different weight factors.

Let $\mathbf{Y}=\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ be a data set consisting of multi-dimensional sensory signals (e.g., acceleration, strain, pressure) from N different run-to-failure units. An ensemble of prognostic member algorithms for RUL prediction can be expressed in a weighted-sum formulation as

$$\hat{L}=\sum_{j=1}^{M} w_j \hat{L}_j(\mathbf{y}_t,\mathbf{Y}) \tag{24}$$

where $\hat{L}$ denotes the ensemble predicted RUL for the testing data set $\mathbf{y}_t$; M denotes the number of algorithm members in the ensemble; w_j denotes the weight assigned to the jth prognostic algorithm; $\hat{L}_j(\mathbf{y}_t, \mathbf{Y})$ denotes the predicted RUL by the jth prognostic member algorithm trained with the data set $\mathbf{Y}$. Let the weight vector $\mathbf{w}=[w_1, \ldots, w_M]^T$ and the vector of predicted RULs by member algorithms $\hat{\mathbf{L}}=[\hat{L}_1,\ldots,\hat{L}_M]^T$, the weighted-sum formulation in Eq. (24) can be expressed in a vector form as $\hat{L}\left(\mathbf{w},\hat{\mathbf{L}}\right)=\mathbf{w}^{\mathrm{T}}\hat{\mathbf{L}}$.

3.2. K-fold cross validation

The k-fold cross validation is used in the offline process to evaluate the accuracy of a given ensemble. It randomly divides the original data set $\mathbf{Y}$ into k mutually exclusive subsets (or folds) $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_k$ having an approximately equal size [38]. Of the k subsets, one is used as the test set and the other $k-1$ subsets are put together as a training set. The CV process is performed k times, with each of the k subsets used exactly once as the test set. Let $\mathbf{I}_m=\{i: \mathbf{y}_i \in \mathbf{Y}_m\}$, $m=1, 2, \ldots, k$ denote the index set of the run-to-failure units whose sensory signals construct the subset $\mathbf{Y}_m$. Then the CV error is computed as the average error over all k trials and can be expressed as

$$\varepsilon_{CV}=\frac{1}{N}\sum_{m=1}^{k}\sum_{i\in \mathbf{I}_m} S(\hat{L}(\mathbf{w},\hat{\mathbf{L}}(\mathbf{y}_i,\mathbf{Y}\backslash\mathbf{Y}_m)),L_i^T) \tag{25}$$

where $S(\bullet)$ is a predefined evaluation metric that measures the accuracy of the ensemble-predicted RUL; N denotes the number of run-to-failure units for CV; Li^T denotes the true RUL of the ith unit. The above formula indicates that all units in the data set are used for both training and testing, and each unit is used for testing exactly once and for training $k-1$ times. Thus, the variance of the resulting estimate is likely to be reduced compared to the traditional holdout approach, resulting in superior performance when employing a small data set. It is important to note that the disadvantage of the k-fold CV against the holdout method is greater computational expense because the training process has to be executed k times. As a commonly used setting for CV, a 10-fold CV is employed in this study.

3.3. Weighting schemes

This section will introduce three schemes to determine the weights of member algorithms: the accuracy-based weighting, diversity-based weighting and optimization-based weighting.

3.3.1. Accuracy-based weighting

The prediction accuracy of the jth member algorithm is quantified by its CV error, expressed as

$$\varepsilon_{CV}^{j}=\frac{1}{N}\sum_{m=1}^{k}\sum_{i\in \mathbf{I}_m} S\left(\hat{L}_j(\mathbf{y}_i,\mathbf{Y}\backslash\mathbf{Y}_m),L_i^T\right) \tag{26}$$

The weight w_j of the jth member algorithm can then be defined as the normalization of the corresponding inverse CV error, expressed as

$$w_j=\frac{(\varepsilon_{CV}^{j})^{-1}}{\sum_{i=1}^{M}(\varepsilon_{CV}^{i})^{-1}} \tag{27}$$

This definition indicates that a larger weight is assigned to a member algorithm with higher prediction accuracy. Thus, a member algorithm with better prediction accuracy has a larger influence on the ensemble prediction. This weighting scheme relies exclusively on the prediction accuracy to determine the weights of member algorithms.

3.3.2. Diversity-based weighting

The weight formulation in Eq. (27) relies exclusively on the prediction accuracy to determine the weights. However, the prediction accuracy of member algorithms is not the only factor that affects the ensemble performance. The prediction diversity, which measures the extent to which the predictions by a member algorithm are distinguishable from those by the others, also has a significant effect on the ensemble performance, especially on the robustness. More specifically, a larger weight should generally be assigned to a member algorithm with higher prediction diversity because of its larger potential to enhance the ensemble robustness.

We begin by formulating an N-dimensional error vector consisting of absolute RUL prediction errors by the jth member algorithm as

$$\mathbf{e}_j = \left[\hat{L}_j(\mathbf{y}_1, \mathbf{Y}\backslash\mathbf{Y}_1) - L_1^T, \ldots, \hat{L}_j(\mathbf{y}_N, \mathbf{Y}\backslash\mathbf{Y}_m) - L_N^T\right]^T \tag{28}$$

Repeatedly computing the error vectors for all M member algorithms gives M error vectors $\mathbf{e}_1$, $\mathbf{e}_2$, …, $\mathbf{e}_M$. The prediction diversity of the jth member algorithm can then be computed as the sum of Euclidean distances between the error vector $\mathbf{e}_j$ and all the other error vectors, given by

$$D_j = \sum_{i=1; i\neq j}^{M} \|\mathbf{e}_j - \mathbf{e}_i\| \tag{29}$$

The prediction diversity measures the extent to which the predictions by a member algorithm are distinguishable from those by any other. Based on the defined prediction diversity, the normalized weight w_j of the jth member algorithm can then be calculated as

$$w_j = \frac{D_j}{\sum_{i=1}^{M} D_i} \tag{30}$$

This definition suggests that a member algorithm with higher prediction diversity will be given a larger weight and thus contributes more to the ensemble predicted RUL. For example, if, among all the member algorithms, one algorithm consistently gives early RUL predictions while any of the others late RUL predictions, the former will likely be given a larger weight than the latter. It is also noted that the weight formulation in Eq. (30) considers the prediction diversity as the only criterion for the weight determination.

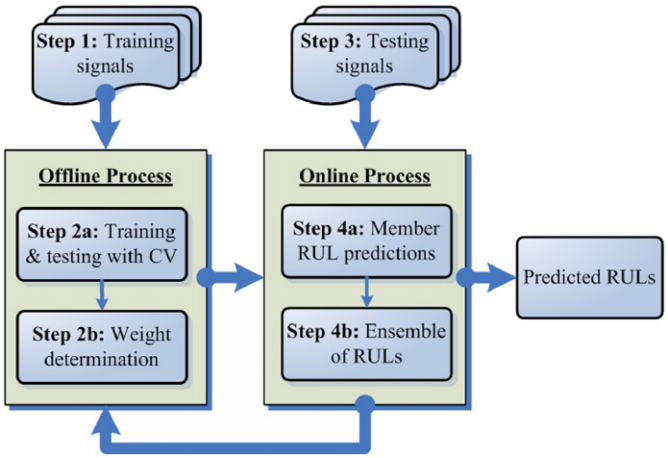

Fig. 2. A flowchart of the ensemble approach.

3.3.3. Optimization-based weighting

Neither the accuracy-based nor diversity-based weighting scheme takes into account both the prediction accuracy and diversity in the weight calculation. Thus, the two schemes cannot produce an ensemble algorithm to achieve both high prediction accuracy and robustness. In what follows, an optimization-based weighting scheme is proposed to maximize the accuracy and robustness of data-driven prognostics by adaptively synthesizing the prediction accuracy and diversity of each member algorithm.

In the optimization-based weighting scheme, the weights in Eq. (24) can be obtained by solving an optimization problem of the following form

$$\begin{aligned} &\text{Minimize } \varepsilon_{CV} = \varepsilon_{CV}(\hat{L}(\mathbf{w}, \hat{\mathbf{L}}(\mathbf{y}_i)), \quad L_i^T, \quad i = 1, \ldots, N) \\ &\text{Subject to } \sum_{j=1}^{M} w_j = 1 \end{aligned} \tag{31}$$

After the prediction of RULs using the M member algorithms through the 10-fold CV, the above optimization problem can be readily solved with almost negligible computational effort since the weight optimization process does not require the execution of member algorithms. Thus, the overall computational cost mainly comes from the training and testing in the CV process. We expect that, by solving the optimization problem in Eq. (31), the resulting ensemble of algorithms will outperform any of the ensemble's individual member algorithms in terms of both accuracy and robustness. The capability of this weighting scheme to adaptively synthesize the prediction accuracy and diversity of each member algorithm will be demonstrated in the case study section.

3.4. Overall procedure

Fig. 2 shows the overall procedure of the proposed ensemble approach with the k-fold CV and three weighting schemes. This data-driven prognostic approach is composed of the offline and online processes. In the offline process, the offline training/testing process with the k-fold CV is employed to compute the CV error of an ensemble formulation; the weights of member algorithms are determined using the accuracy-based weighting, diversity-based weighting and optimization-based weighting. The online prediction process combines the RUL predictions from all member algorithms to form an ensemble RUL prediction using the weights obtained from the offline process. This process enables the continuous update of the health information and prognostic results in real-time with new sensory signals. Table 2 details the proposed ensemble prognostics approach with the five steps. STEPS 2–4 can be repeated to incorporate new training sensory signals and to update the weights and RUL predictions. Since the computationally expensive training process with multiple algorithms is done offline and the online prediction process with multiple algorithms requires a relatively small amount of computational effort, the ensemble approach raises little concerns in the computational complexity. Indeed, in many engineered systems, the prognostic accuracy is treated as of much higher

Table 2
Detailed procedure of the ensemble approach.

STEP 1	Determine sensor configurations and acquire training sensory signals from offline system units.
STEP 2a	Perform the offline training and testing processes with the k-fold CV with the training sensory signals to compute the CV error.
STEP 2b	Determine the weights using the accuracy-based weighting, diversity-based weighting and optimization-based weighting schemes.
STEP 3	Acquire testing sensory signals from online system units.
STEP 4a	Predict RULs using the member algorithms through the online prediction process which employs the background health knowledge obtained from the offline training process.
STEP 4b	Predict the ensemble RULs with the optimum weights obtained from STEP 2b.

importance compared to the computational complexity since the occurrence of a catastrophic system failure causes much more loss than the increase of the computational efforts. Therefore, in cases where the ensemble approach achieves considerable improvement in the prediction accuracy over any sole member algorithm, we should always prefer the use of the former.

4. Case studies

In this section, the proposed ensemble of data-driven prognostic algorithms is demonstrated with three PHM case studies: (i) 2008 IEEE PHM challenge problem, (ii) power transformer problem, and (iii) electric cooling fan problem. In each case study, the ensemble approach combines RUL predictions from five popular data-driven prognostic algorithms, namely, a similarity-based interpolation (SBI) approach with RVM as the regression technique (RVM–SBI) [15,49], SBI with SVM (SVM–SBI) [47,15], SBI with the least-square exponential fitting (Exp–SBI) [15], a Bayesian linear regression with the least-square quadratic fitting (Quad–BLR) [17], and a recurrent neural network (RNN) approach [50,18]. Details regarding the five prognostic algorithms are given in Section 2.

4.1. 2008 IEEE PHM challenge problem

In an aerospace system (e.g., an airplane, a space shuttle), system safety plays an important role since failures can lead to dramatic consequences. In order to meet stringent safety requirements as well as minimize the maintenance cost, condition-based maintenance must be conducted throughout the system's lifetime, which can be enabled by system health prognostics. This case study aims at predicting the RULs of aircraft engine systems in an accurate and robust manner with massive and heterogeneous sensory data.

4.1.1. Description of data set

The data set provided by the 2008 IEEE PHM Challenge problem consists of multivariate time series signals that are collected from an engine dynamic simulation process. Each time series signal comes from a different degradation instance of the dynamic simulation of the same engine system [39]. The data for each cycle of each unit include the unit ID, cycle index, 3 values for an operational setting and 21 values for 21 sensor measurements. The sensor data were contaminated with measurement noise and different engine units start with different initial health conditions and manufacturing variations which are unknown. Three operational settings have a substantial effect on engine degradation behaviors and result in six different operation regimes as shown in Table 3. The 21 sensory signals were obtained from 6 different operation regimes. The whole data set was divided into training and testing subsets, each of which consists of 218 engine units. In the training data set, the damage growth in a unit was allowed until the occurrence of a system failure when one or more limits for safe operation have been reached. In the testing data set, the time series signals were pruned some time prior to the occurrence of a system failure. The objective of the problem is to predict the number of remaining operational cycles before failure in the testing data set.

4.1.2. Implementation of ensemble approach

For the CV process, the training data set with 218 units were divided to 10 data subsets with a similar size. Each data subset was used for both training and testing and, more specifically, nine times for training and once for testing. The training data subsets contain complete degradation information while the testing data subsets carry only partial degradation information. The latter were generated by truncating the original data subsets after pre-assigned RULs. The RUL pre-assigned to each unit in a testing data subset was randomly generated from a uniform distribution between its zero and half-remaining life. This range in the uniform distribution was selected based on the following two criteria: (i) the pre-assigned RULs should be small enough to allow the occurrence of substantial degradation; and (ii) the variation of the pre-assigned RULs should be large enough to test the robustness of algorithms.

Table 3
Six different operation regimes.

Regime ID	Operating parameter 1	Operating parameter 2	Operating parameter 3
1	0	0	100
2	20	0.25	20
3	20	0.7	0
4	25	0.62	80
5	35	0.84	60
6	42	0.84	40

Among the 21 sensory signals, some signals contain no or little degradation information of an engine unit whereas the others do. To improve the RUL prediction accuracy, important sensory signals must be carefully selected to characterize the degradation behavior of engine units for health prognostics. Following the work in [15], this study selected 7 sensory signals (2, 3, 4, 7, 11, 12 and 15) among the 21 sensory signals for the use in the member algorithms: RVM–SBI, SVM–SBI, Exp–SBI and Quad–BLR. A monotonic lifetime trend can be observed from these seven sensory signals of which the noise levels are relatively low. For the VHI construction, the system failure matrix $\mathbf{Q}_0$ was created with the sensory data in a system failure condition, $0 \le L \le 4$, while the system healthy matrix $\mathbf{Q}_1$ with those in a system healthy condition, $L > 300$. The RVM employed a linear spline kernel function with the initial most probable hyper-parameter vector for kernel weights $\boldsymbol{\alpha}_m = [1 \times 10^4, \ldots, 1 \times 10^4]$ and the initial most probable noise variance $\sigma_m^2 = 1 \times 10^{-4}$. In the SVM, a Gaussian kernel function is used with the parameter settings as: the regularization parameter $C = 10$ and the parameter of the ε-insensitive loss function $\varepsilon = 0.10$. In the RNN training, the 21 normalized sensory signals together with the regime ID at each cycle were used as the multi-dimensional inputs of the RNN and the RUL at the corresponding cycle was used as the output. The implementation details can be found in [18]. In the RNN architecture, the numbers of the input, recurrent and output units are $|I| = 22$, $|R| = 8$ and $|O| = 1$.

The evaluation metric considered for this example employed an asymmetric score function around the true RUL such that heavier penalties are placed on late predictions [39]. The score evaluation metric S can be expressed as

$$S(\hat{L}_i, L_i^T) = \begin{cases} \exp(-d_i/13) - 1, & d_i < 0 \quad \text{where} \quad d_i = \hat{L}_i - L_i^T \\ \exp(d_i/10) - 1, & d_i \ge 0 \end{cases} \tag{32}$$

where $\hat{L}_i$ and L_i^T denote the predicted and true RUL of the ith unit, respectively. This score function was used to compute the CV error ε_{CV} using Eq. (25) for the accuracy- and optimization-based weighting schemes. In this study the weight optimization problem in Eq. (31) was solved using a sequential quadratic optimization (SQP) method which is a gradient-based optimization technique.

Table 4
Weighting results, CV and validation errors for 2008 PHM challenge problem.

	RS	SS	ES	QB	RN	RS–SS–ES–QB–RN		
						AW	DW	OW
Weight by AW	0.3063	0.3029	0.3137	0.0151	0.0620	–	–	–
Weight by DW	0.1478	0.1488	0.1488	0.3354	0.2191	–	–	–
Weight by OW	0.0000	0.0470	0.7462	0.2068	0.0000	–	–	–
CV error	8.0743	8.1646	7.8834	163.3376	39.8583	6.9159	7.0852	**4.8387**
Validation error	10.2393	9.3907	10.4710	247.0079	20.1499	8.5544	**6.1280**	6.1955

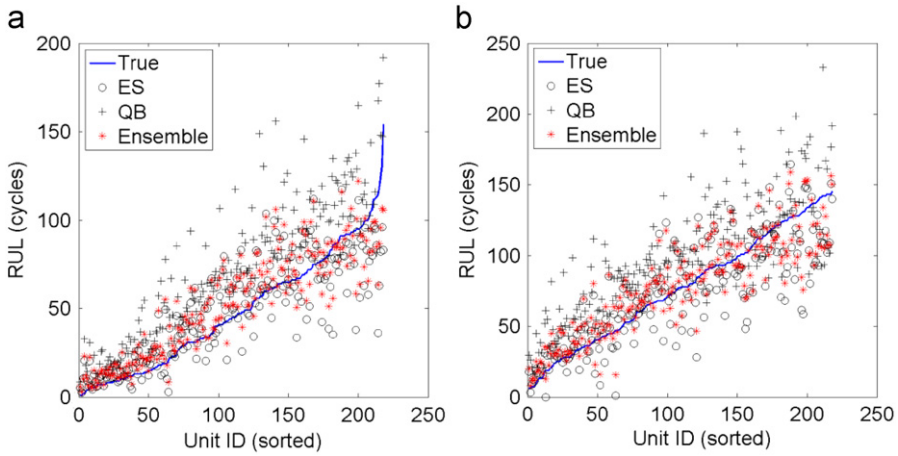

Fig. 3. RUL predictions of training units (a) and testing units (b) for 2008 PHM challenge problem (optimization-based weighting).

4.1.3. Results of ensemble approach

The five selected member algorithms are RVM–SBI (RS), SVM–SBI (SS), Exp–SBI (ES), Quad–BLR (QB) and RNN (RN). The three weighting schemes are the accuracy-based weighting (AW), diversity-based weighting (DW) and optimization-based weighting (OW). Table 4 summarizes the weighting results by the three weighting schemes as well as compares the CV and validation errors of the individual and ensemble approaches. It is observed that the ensemble approaches with all three weighting schemes outperforms any of the individual member algorithm in terms of the CV error and that the one with the optimization-based weighting achieves the smallest CV error of 4.8387 on the training data set, a 38.62% improvement over the best individual member algorithm, ES, whose CV error is 7.8834. As expected, the accuracy-based weighting scheme yields better prediction accuracy than the diversity-based weighting. This can be attributed to the fact that the former assigns larger weights to member algorithms with better prediction accuracy while the latter does not consider the prediction accuracy in the weight determination. To test the robustness of the ensemble approaches, the testing data set with 218 units were employed to compute the validation errors. Note that the testing data set is different from the training data set that was used to determine the weights in the ensemble approach. It is apparent that the ensemble approaches again outperform the individual member algorithms and that the one with the diversity-based weighting performs best, with a 34.7% improvement over the best individual member algorithm, SS. This suggests that the diversity-based weighting, compared to the accuracy-based weighting, provides a more robust ensemble of the member algorithms. It is noted that the optimization-based weighting scheme still achieves a comparable validation error to that of the diversity-based weighting scheme.

Under the optimization-based weighting scheme, the RUL predictions by two individual algorithms, ES and QB, with the largest weights and the ensemble approach are plotted for 218 training and testing units in Fig. 3. The units are sorted by the RULs in an ascending order. It is seen that ES tends to give consistently early RUL predictions while QB tends to provide consistently late RUL predictions. In contrast, the ensemble approach gives RUL predictions closer to the true values while eliminating many outliers produced by the two individual algorithms. The optimization-based weighting scheme provides better performance since the scheme employs an optimum ensemble formulation.

4.1.4. Comparison of different combinations of member algorithms

Out of the five member algorithms, 31 different combinations can be chosen to formulate an ensemble approach. It would be interesting to study how a choice of combination affects the performance of an ensemble approach. Table 5 summarizes the CV errors for ensemble approaches with all possible combinations of the member algorithms under the optimization-based weighting scheme. Three important remarks can be derived from the results. First of all, it is observed that the ES, as the individual member algorithm with the best performance, always serves as a member algorithm of the best ensemble approach. We also observe that the ES, when involved in the ensemble approach, always had a larger weight than any other. It indicates that the best member algorithm exhibits good cooperative performance which can be identified by the optimization-based weighting scheme. Second, the QB, which gives the worst individual performance, was surprisingly selected as an important member of the best ensemble approach. These results, though counterintuitive, suggest that the ensemble approach can adaptively synthesize the

Table 5
Comparison of CV errors of different combinations of member algorithms for 2008 PHM challenge problem (optimization-based weighting).

Combination	CV error	Combination	CV error	Combination	CV error
RS	8.0743	RS–SS	8.0769	RS–SS–ES	7.8834
SS	8.1646	RS–ES	7.8834	RS–SS–QB	4.9123
ES	**7.8834**	RS–QB	4.9162	RS–SS–RN	6.7983
QB	163.3376	RS–RN	6.8002	RS–ES–QB	4.8391
RN	39.8583	SS–ES	7.8834	RS–ES–RN	6.5194
Mean	*45.4636*				
Std[a]	*67.3188*				
RS–SS–ES–QB	**4.8387**	SS–QB	4.9362	RS–QB–RN	4.9162
RS–SS–ES–RN	6.5194	SS–RN	6.8376	**SS–ES–QB**	**4.8387**
RS–SS–QB–RN	4.9123	**ES–QB**	**4.8391**	SS–ES–RN	6.5194
RS–ES–QB–RN	4.8391	ES–RN	6.5194	SS–QB–RN	4.9362
SS–ES–QB–RN	**4.8387**	QB–RN	17.5868	ES–QB–RN	4.8391
Mean	*5.1896*	*Mean*	*7.6279*	*Mean*	*5.7002*
Std	*0.7440*	*Std*	*3.7182*	*Std*	*1.1234*
RS–SS–ES–QB–RN	4.8387				

[a] Standard deviation

prediction ability and diversity of each individual algorithm to enhance the accuracy and robustness of RUL predictions. Indeed, the QB is prone to give late RUL predictions as shown in Fig. 3 and thus possesses higher prediction diversity. Third, both the mean and standard deviation of CV errors decrease as the number of member algorithms increases. The mean and standard deviation of CV errors of ensemble approaches with a single member algorithm are 45.4636 and 67.3188, respectively, and they monotonically decrease to 5.1896 and 0.7440, respectively, by the ensemble approach with four member algorithms. Thus it would be beneficial to have more member algorithms to enhance the prediction accuracy and reduce the uncertainty of this accuracy.

4.2. Power transformer problem

The power transformer is a critical power element in nuclear power plants, since an unexpected breakdown of the transformer causes plant shut-down and substantial societal expense. So it is very important to ensure high reliability and safety of the transformer during its operation. Investigations of the failures causes have revealed that mechanical breakdowns constitute a large portion of unexpected breakdowns of transformers in nuclear power plants [40]. Therefore, health monitoring and prognostics of the transformer with respect to mechanical failures is of significant importance to preventing unexpected breakdowns and minimizing interruptions to reliable customer service. This case study conducts transformer health prognostics with sensory signals obtained from a finite element (FE) model of a power transformer.

4.2.1. Model description

The FE model of a power transformer was created in ANSYS 10 as shown in Fig. 4, where one exterior wall is concealed to make the interior structure visible. The transformer is fixed at the bottom surface and a vibration load with the frequency of 120 Hz is applied to the magnetic core. The three windings have a total number of twelve support joints, with each having four support joints. The random parameters considered in this study are listed in Table 6, which includes the material properties of support joints and windings as well the geometries of the transformer. The uncertainties in vibration responses propagated from these uncertain parameters will be accounted for when generating prognostic data.

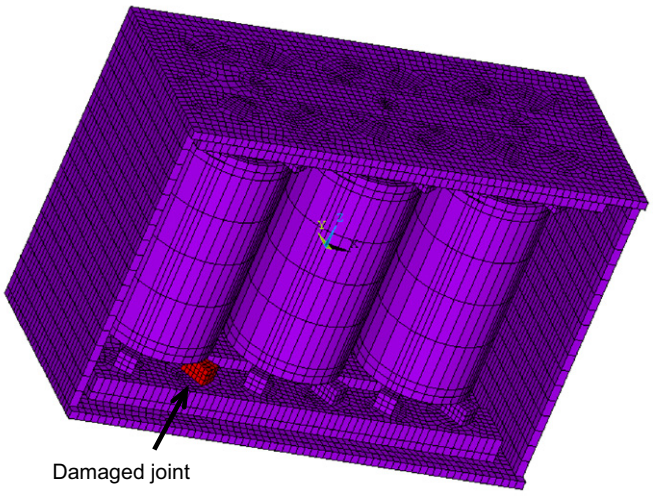

Fig. 4. A power transformer FE model (without the covering wall).

Table 6
Random geometries and material properties for power transformer problem.

Component	Physical meaning	Distri. type	Mean	Std
x_1	Wall thickness	Normal	3	0.015
x_2	Angular width of support joints	Normal	15	0.075
x_3	Height of support joints	Normal	6	0.03
x_4	Young's modulus of support joint	Normal	2E+12	1E+10
x_5	Young's modulus of winding	Normal	1.28E+12	6E+8
x_6	Poisson ratio of joints	Normal	0.27	0.0027
x_7	Poisson ratio of winding	Normal	0.34	0.0034
x_8	Density of joints	Normal	7.85	0.000785
x_9	Density of windings	Normal	8.96	0.0896

Since it is very difficult, if not impossible, to obtain direct measurements of the health condition of transformers, indirect measurements are most often used to diagnose the health condition and predict the RULs of transformers [41]. In particular, the vibrations of the magnetic core and of the windings could characterize transitory overloads and permanent failures before any irreparable damage occurs [42,43]. Thus, this case study employs the vibration signals of the magnetic core and of the windings of a power transformer to predict the RULs of transformers.

4.2.2. Prognostic data generation

The failure mode considered in this study is the loosening of a winding support joint (see Fig. 4) induced by the magnetic core vibration. The joint loosening was realized by reducing the stiffness of the joint. The failure criterion is defined as a 99% stiffness reduction of the joint. To model the trajectory of change in stiffness over time, this study uses a damage propagation model with an exponential form as [39]

$$E(t) = E_0 + b_E(1 - \exp(a_E t)) \tag{33}$$

where E_0 is the initial Young's modulus of the joint; a_E and b_E are the model parameters; t is the cycle time. The initial Young's modulus E_0 follows the same normal distribution with x_4 (see Table 6). The model parameters a_E and b_E are independent and normally distributed with means 0.002 and 4E+12, each of which has a 10% coefficient of variation.

Since data-driven prognostic approaches require a large amount of prognostic data, it is computationally intolerable, if

not impossible, to simply run the simulation to generate every data point. To overcome this difficulty, this study employed the univariate decomposition method that only uses a certain number of univariate sample points to construct the response surface for a general multivariate response function while achieving good accuracy [44]. This study selected five strain gauges (see Fig. 5) from the optimally designed sensor network consisting of nine strain gages and thus requires the construction of five responses surfaces. The data generation process involves four sequentially executed procedures: (i) Four univariate sample points were obtained from the harmonic analysis to construct response surfaces, along the damage propagation path, that approximate the strain components at five sensor locations as functions of random variables detailed in Table 6; (ii) 400 randomly generated samples of E_0, a_E and b_E were used in conjunction with Eq. (33) to produce 400 damage propagation paths, of which 200 paths were assigned to the training units and the rest to the testing units; (iii) the constructed response surfaces were used to interpolate the strain components at five sensor locations for a given set of randomly generated geometries and material properties and damage propagation paths, and repeatedly executing this process for 400 times gave the training data set with 200 training units and the testing data set with 200 testing units; (iv) measurement noise following a zero mean normal distribution was added to both the training and testing data sets to finalize the data generation. The cubic spline was used as the numerical scheme for the response surface construction and interpolation. Simulated measurements by sensors 1 and 5 are plotted against the adjusted cycle index, defined as the subtraction of the cycle-to-failure from the actual operational cycle, in Fig. 6 for all 200 training units in the training data set.

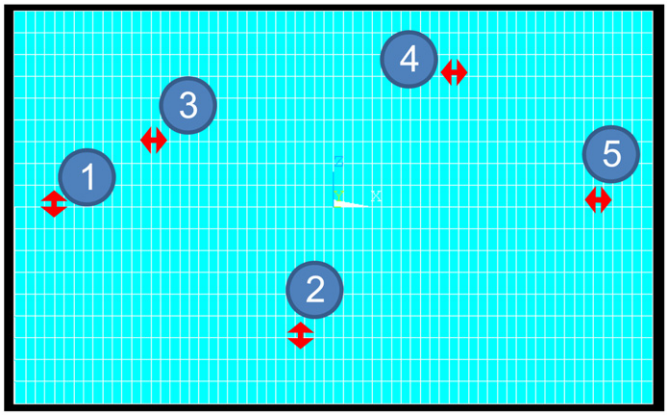

Fig. 5. strain gauges located on the side wall of power transformer.

4.2.3. Implementation of ensemble approach

The training data set with 200 units were equally and randomly divided to 10 subsets. Similar to the first example, when used for the testing in CV, each unit in a subset was assigned with a randomly generated RUL from a uniform distribution between its zero and half-remaining life. All the five member algorithms used the same parameter settings with those detailed in Section 4.1.2. The score function in Eq. (32) was again used to compute the CV error ε_{CV} for the accuracy- and optimization-based weighting schemes.

4.2.4. Results of ensemble approach

Table 7 summarizes the weighting results by the three weighting schemes as well as compares the CV and validation errors of the individual and ensemble approaches. Compared to the first example, similar results can be observed: (i) The ensemble approaches with all three weighting schemes yield smaller CV error than any of the individual member algorithm and the one with the optimization-based weighting gives the smallest CV error of 2.7258 on the training data set, a 66.48% improvement over the best individual member algorithm, RN, whose CV error is 8.1323; (ii) the accuracy-based weighting scheme yields a comparable CV error to that of the diversity-based weighting; (iii) the optimization-based weighting scheme achieves a validation error of 5.6138, which is comparable to the smallest validation error of 5.6119 by the diversity-based weighting scheme.

Under the optimization-based weighting scheme, the RUL predictions by two individual algorithms, ES and QB, with the largest weights and the ensemble approach are plotted for 218 training and testing units in Fig. 7. It can be observed that ES and QB are prone to produce early and late RUL predictions, respectively, while the ensemble approach gives RUL predictions closer to the true values with a much smaller number of outliers.

4.2.5. Comparison of different combinations of member algorithms

A comparison study of different combinations of member algorithms was again carried out using the optimization-based weighting scheme for the power transformer problem. Table 8 summarizes the comparison results from which several important remarks similar to those in the first example can be derived. First of all, the member algorithms ES and QB can always be observed in the best ensemble approach with more than one member

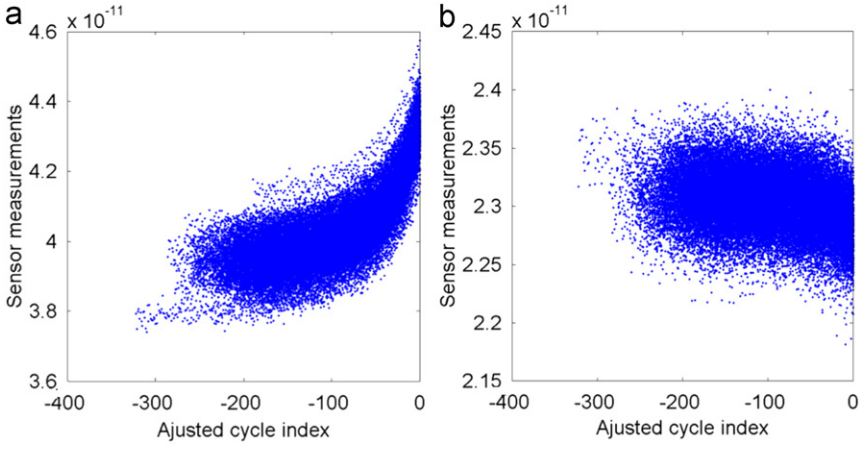

Fig. 6. Simulated measurements by sensors 1(a) and 5(b) for power transformer problem.

Table 7
Weighting results, CV and validation errors for power transformer problem.

	RS	SS	ES	QB	RN	RS–SS–ES–QB–RN		
						AW	DW	OW
Weight by AW	0.2128	0.2265	0.2343	0.0677	0.2588	–	–	–
Weight by DW	0.1488	0.1486	0.1688	0.3290	0.2048	–	–	–
Weight by OW	0.0000	0.0000	0.6303	0.2336	0.1361	–	–	–
CV error	9.8922	9.2945	8.9849	31.0891	8.1323	3.4874	3.4124	**2.7258**
Validation error	6.5737	6.8847	7.8251	20.0356	15.2265	5.7825	**5.6119**	5.6138

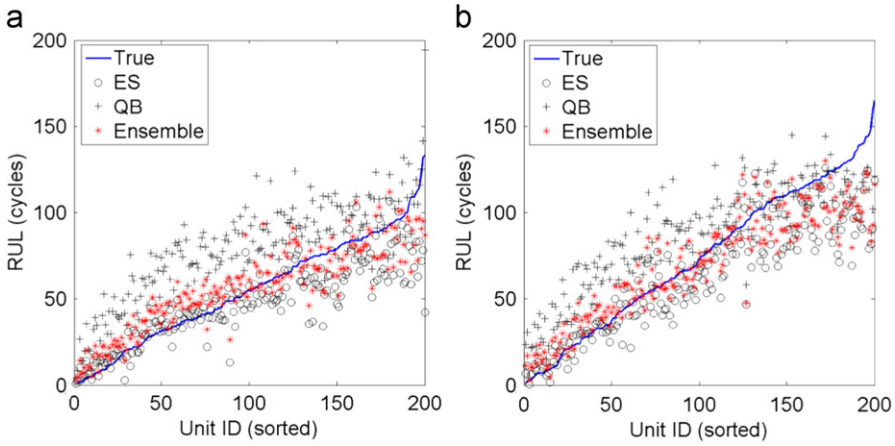

Fig. 7. RUL predictions of training units (a) and testing units (b) for power transformer problem (optimization-based weighting).

algorithms. We also observe that the combination ES and QB, when involved in the ensemble approach, always had a larger weight than any other. This result is different from what we observe in the first example, where the largest weight was assigned to the best individual member algorithm. This suggests that the optimization-based weighting scheme makes less use of or even discarded the best member algorithm that does not exhibit good cooperative performance with other members. Second, the QB, which gives the worst individual performance and is prone to give later RUL predictions (see Fig. 7), was selected as an important member of the best ensemble approach. This again suggests that the prediction diversity plays an important role in the weight determination. Third, as is the case in the first example, both the mean and standard deviation of CV errors decrease as the number of member algorithms increases. Thus the addition of member algorithms tends to enhance the prediction accuracy and reduce the uncertainty of this accuracy.

4.3. Electric cooling fan problem

In addition to the numerical studies, we also conducted experimental testing to verify the effectiveness of the ensemble approach. In this case study, we applied the ensemble approach to the health prognostics of electronic cooling fan units. Cooling fans are one of the most critical parts in system thermal solution of most electronic products [51] and in cooling towers of many chemical plants [52]. This study aims to demonstrate the proposed ensemble prognostics with 32 electronic cooling fans.

4.3.1. Experimental setup

In this experimental study, thermocouples and accelerometers were used to measure temperature and vibration signals. To make time-to-failure testing affordable, the accelerated testing condition

Table 8
Comparison of CV errors of different combinations of member algorithms for power transformer problem (optimization-based weighting).

Combination	CV error	Combination	CV error	Combination	CV error
RS	9.8922	RS–SS	9.2945	RS–SS–ES	8.9561
SS	9.2945	RS–ES	8.9849	RS–SS–QB	3.1688
ES	8.9849	RS–QB	3.1764	RS–SS–RN	3.9651
QB	31.0891	RS–RN	3.9744	RS–ES–QB	2.7894
RN	**8.1323**	SS–ES	8.9561	RS–ES–RN	3.4557
Mean	*13.4786*				
Std	*9.8650*				
RS–SS–ES–QB	2.7894	SS–QB	3.1815	RS–QB–RN	3.1470
RS–SS–ES–RN	3.4557	SS–RN	3.9671	SS–ES–QB	2.7894
RS–SS–QB–RN	3.1433	**ES–QB**	**2.7894**	SS–ES–RN	3.4557
RS–ES–QB–RN	**2.7258**	ES–RN	3.4557	SS–QB–RN	3.1559
SS–ES–QB–RN	**2.7258**	QB–RN	6.9724	**ES–QB–RN**	**2.7258**
Mean	*2.9680*	*Mean*	*5.4752*	*Mean*	*3.7609*
Std	*0.3232*	*Std*	*2.7412*	*Std*	*1.8640*
RS–SS–ES–QB–RN	2.7258				

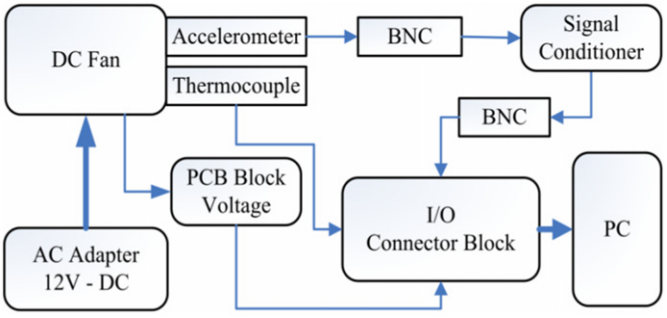

Fig. 8. DC fan degradation test block diagram.

for the DC fan units was sought with inclusion of a small amount of tiny metal particles into ball bearings and an unbalanced weight on one of the fan units. The experiment block diagram of DC fan accelerated degradation test is shown in Fig. 8. As shown in the diagram, the DC fan units were tested with 12 V regulated power supply and three different signals were measured and stored in a PC through a data acquisition system. Fig. 9(a) shows the test fixture with four screws at each corner for the DC fan units. As shown in Fig. 9(b), an unbalanced weight was used and mounted on one blade for each fan. Sensors were installed at different parts of the fan, as shown in Fig. 10. In this study, three different signals were measured: the fan vibration signal by the accelerometer, the Printed Circuit Board (PCB) block voltage by the voltmeter, and the temperature measured by the thermocouple. An accelerometer was mounted to the bottom of the fan with superglue, as shown in Fig. 10(a). Two wires were connected to the PCB block of the fan to measure the voltage between two fixed points, as shown in Fig. 10(b). As shown in Fig. 10(c), a thermocouple was attached to the bottom of the fan and measures the temperature signal of the fan. Vibration, voltage, and temperature signals were acquired by the data acquisition system and stored in PC. The data acquisition system from National Instruments Corp. (NI USB 6009) and the signal conditioner from PCB Group, Inc. (PCB 482A18) were used for the data acquisition system. In total, 32 DC fan units were tested at the same condition and all fan units run till failure.

4.3.2. Implementation of ensemble approach

The sensory signal screening found that the fan PCB block voltage and the fan temperature did not show clear degradation trend, whereas the vibration signal showed health degradation behavior. This study involved the root mean squares (RMS) of the vibration spectral responses at the first five resonance frequencies and defined the RMS of the spectral responses as the PHI for the DC fan prognostics. Fig. 11 shows the RMS signals of three fan units to demonstrate the health degradation behavior. The RMS signal gradually increased as the bearing in the fan degraded over time. It was found that the PHI is highly random and non-monotonic because of metal particles, sensory signal noise, and input voltage noise.

Among 32 fan units, the first 20 fan units were used to construct the training data set for the CV, while the rest were used to build the testing data set for the validation. Due to the small amount of training data, this case study employed the 5-fold CV where the training data set with 20 units was equally and randomly divided to five subsets. Similar to the previous

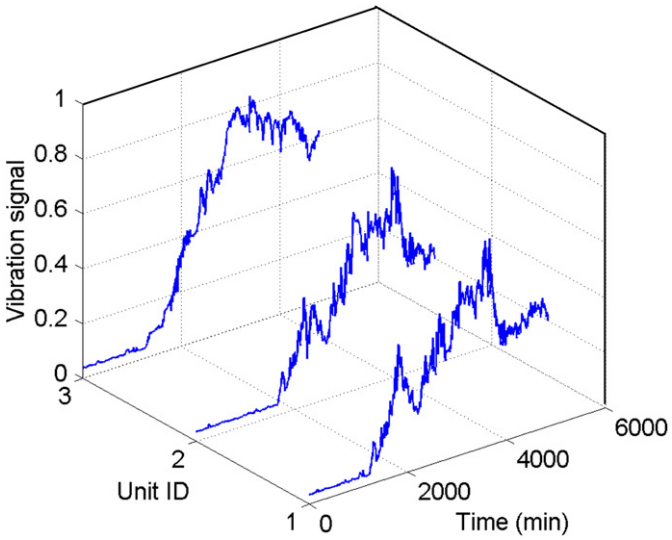

Fig. 11. Sample degradation signals from DC fan testing.

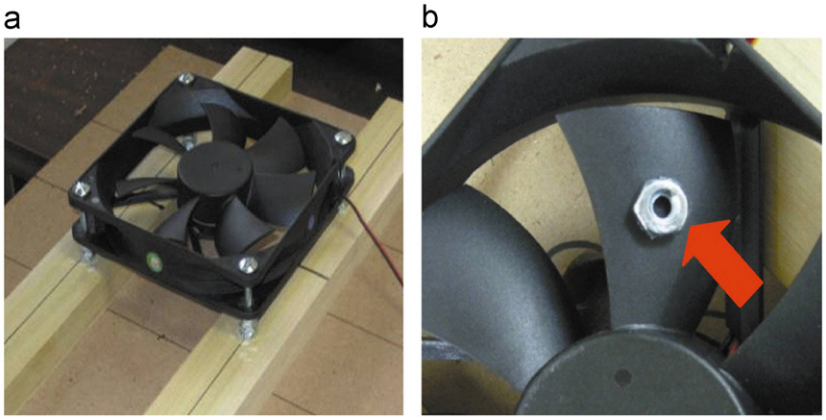

Fig. 9. DC fan test fixture (a) and the unbalance weight installation (b).

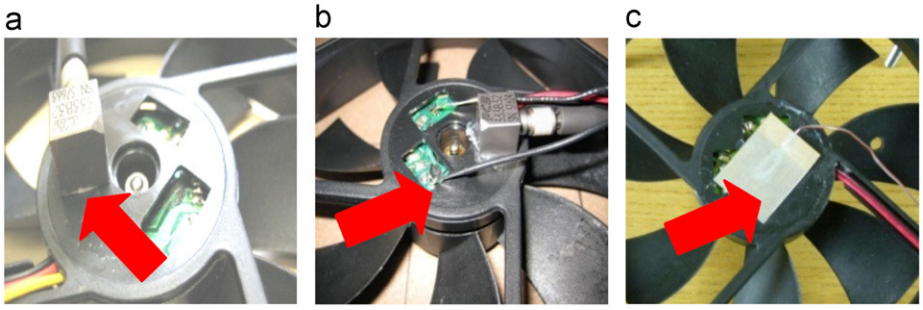

Fig. 10. Sensor installations for DC fan test: (a) accelerometer, (b) voltmeter and (c) thermocouples.

Table 9
Weighting results, CV and validation errors for electric cooling fan problem.

	RS	SS	ES	QB	RN	RS–SS–ES–QB–RN		
						AW	DW	OW
Weight by AW	0.3646	0.3767	0.2552	0.0008	0.0027	–	–	–
Weight by DW	0.1423	0.1427	0.1496	0.3285	0.2369	–	–	–
Weight by OW	0.1155	0.8845	0.0000	0.0000	0.0000	–	–	–
CV error	1.4770	1.4298	2.1100	717.8430	199.0067	1.5188	11.8520	**1.4292**
Validation error	0.7027	0.9223	0.7037	461.5064	84.3975	0.7185	11.0177	**0.6984**

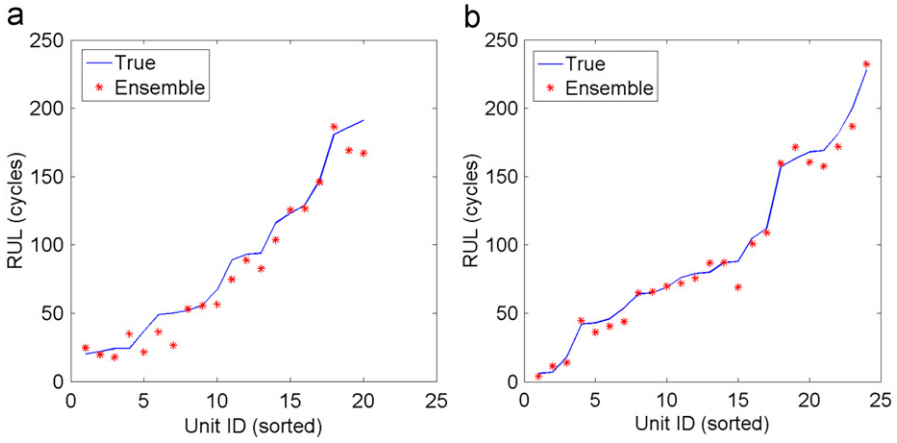

Fig. 12. RUL predictions of training units (a) and testing units (b) for electric cooling fan problem (optimization-based weighting).

examples, when used for the testing in CV, each unit in a subset was assigned with a randomly generated RUL from a uniform distribution between its zero and half-remaining life. To expand the number of testing units, each testing fan unit was assigned with two randomly generated RUL from a uniform distribution between its zero and half-remaining life, resulting in totally 24 testing units. The parameter settings detailed in Section 4.1.2 was again used for the five member algorithms. With one cycle defined as every ten minutes, the score function in Eq. (32) was again used to compute the CV error ε_{CV} for the accuracy- and optimization-based weighting schemes.

4.3.3. *Results of ensemble approach*

The weighting results by the three weighting schemes and the CV and validation errors of the individual and ensemble approaches are summarized in Table 9. Compared to the previous examples, we observed quite different results from which three important remarks can be derived. First of all, the ensemble approach with the diversity-based weighting scheme gives considerably larger CV and validation errors than the best individual member algorithms, RS and ES. This result is due to the fact that the diversity-based weighting, which relies exclusively on the prediction diversity for the weight determination, assigned larger weights to the member algorithms, QB and RN, which produced very low prediction accuracy due to the random and non-monotonic nature of the PHI (see Fig. 11). Second, compared to the best individual member algorithms, RS and SS, the ensemble approach with the optimization-based weighting gave smaller CV and validation errors. However, the improvement is insignificant. Since non-zero weights are only assigned to the two member algorithms, RS and ES, with superb prediction capability, the performance of the resulting ensemble is totally determined by these two algorithms. However, RS and ES gave similar RUL predictions and the resulting ensemble, which is indeed a combination of two algorithms with similar prediction behavior, cannot achieve significant improvement in the prediction performance. Therefore, we expect that the ensemble approach achieves significant improvement in the prediction performance only in cases where member algorithms with comparable prediction accuracy produce diverse RUL predictions. Third, although the member algorithms, QB and RN, have larger prediction diversity, their prediction accuracy is not comparable with that of the best member algorithms, RS and SS. As a result, these two algorithms were discarded from the algorithm pool by the optimization-based weighting. Under the optimization-based weighting scheme, the RUL predictions by the ensemble approach are plotted for the training and testing units in Fig. 12 where we observed very accurate RUL predictions by the ensemble approach.

5. Conclusion

This paper proposed a novel ensemble approach for the data-driven prognostics of high-risk engineered systems. By combining the predictions of all member algorithms, the ensemble approach achieves better accuracy in RUL predictions compared to any sole member algorithm. Furthermore, the ensemble approach has an inherent flexibility to incorporate any advanced prognostic algorithm that will be newly developed. To the best of our knowledge, this is the first study of an ensemble approach with three weighting scheme for the data-driven prognostics. Since the computationally expensive training process is done offline and the online prediction process requires a small amount of computational effort, the ensemble approach raises little concerns in the computational feasibility. Three engineering case studies (2008 PHM challenge problem, power transformer problem and electric cooling fan

problem) demonstrated the superb performance of the proposed ensemble approach for the data-driven prognostics. Among the three weighting schemes, the optimization-based weighting scheme showed the capability of adaptively synthesizing the prediction accuracy and diversity of each member algorithm to enhance the accuracy of RUL predictions. Considering the enhanced accuracy and robustness in RUL predictions, the proposed ensemble approach leads to the possibility of effective condition-based maintenance practice and risk-informed lifetime management of high-risk engineered systems.

Acknowledgement

This work was partially supported by a grant from the Energy Technology Development Program (2010101010027B) and International Collaborative R&D Program (0420-2011-0161) of Korea Institute of Energy Technology Evaluation and Planning (KETEP), funded by the Korean government's Ministry of Knowledge Economy, the National Research Foundation of Korea (NRF) grant (No. 2011-0022051) funded by the Korea government, the Basic Research Project of Korea Institute of Machinery and Materials (Project Code: SC0830) supported by a grant from Korea Research Council for Industrial Science & Technology, and the Institute of Advanced Machinery and Design at Seoul National University (SNU-IAMD).

References

[1] Dekker R. Applications of maintenance optimization models: a review and analysis. Reliability Engineering and System Safety 1996;51(3):229–40.
[2] Marseguerra M, Zio E, Podofillini L. Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation. Reliability Engineering and System Safety 2002;77(2):151–65.
[3] Zio E. Review reliability engineering: old problems and new challenges. Reliability Engineering and System Safety 2009;94(2):125–41.
[4] Barata J, Guedes Soares C, Marseguerra M, Zio E. Simulation modelling of repairable multi-component dteriorating systems for 'on condition' maintenance optimization. Reliability Engineering and System Safety 2002;76(3): 255–64.
[5] Grall A, Berenguer C, Dieulle L. A condition-based maintenance policy for stochastically deteriorating systems. Reliability Engineering and System Safety 2002;76(3):167–80.
[6] Weide JAM, van der, Pandey MD, van Noortwijk JM. Discounted cost model for condition-based maintenance optimization. Reliability Engineering and System Safety 2010;95(3):236–46.
[7] de Smidt-Destombes KS, van der Heijden MC, van Harten A. On the availability of a k-out-of-N system given limited spares and repair capacity under a condition based maintenance strategy. Reliability Engineering and System Safety 2004;83(3):287–300.
[8] Tinga T. Application of physical failure models to enable usage and load based maintenance. Reliability Engineering and System Safety 2010;95(10): 1061–75.
[9] Myotyri E, Pulkkinen U, Simola K. Application of stochastic filtering for lifetime prediction. Reliability Engineering and System Safety 2006;91(2): 200–8.
[10] Cadini F, Zio E, Avram D. Model-based Monte Carlo state estimation for condition-based component replacement. Reliability Engineering and System Safety 2009;94(3):752–8.
[11] Luo J, Pattipati KR, Qiao L, Chigusa S. Model-based prognostic techniques applied to a suspension system. IEEE Transactions on Systems, Man and Cybernetics, Part A 2008;38(5):1156–68.
[12] Gebraeel N, Pan J. Prognostic degradation models for computing and updating residual life distributions in a time-varying environment. IEEE Transactions on Reliability 2008;57(4):539–50.
[13] Gebraeel N, Elwany A, Pan J. Residual life predictions in the absence of prior degradation knowledge. IEEE Transactions on Reliability 2009;58(1):106–17.
[14] Schwabacher M., 2005, A survey of data-driven prognostics, Proceedings of AIAA Infotech@Aerospace Conference, Arlington, VA.
[15] Wang T, Yu J, Siegel D, and Lee J., 2008, A similarity-based prognostics approach for remaining useful life estimation of engineered systems, International Conference on Prognostics and Health Management, Denver, CO, Oct 6–9.
[16] Zio E, Di Maio F. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear power plant. Reliability Engineering and System Safety 2010;95(1):49–57.
[17] Coble JB, and Hines JW., 2008, Prognostic algorithm categorization with PHM challenge application, IEEE International Conference on Prognostics and Health Management, Denver, CO, Oct 6–9.
[18] Heimes FO., 2008, Recurrent neural networks for remaining useful life estimation, IEEE International Conference on Prognostics and Health Management, Denver, CO, Oct 6–9.
[19] Kozlowski JD, Watson MJ, Byington CS, Garga AK, and Hay TA., 2001, Electrochemical cell diagnostics using online impedance measurement, state estimation and data fusion techniques, Proceedings of 36th Intersociety Energy Conversion Engineering Conference, Savannah, GA.
[20] Goebel K, Eklund N, and Bonanni P., 2006, Fusing competing prediction algorithms for prognostics, Proceedings of 2006 IEEE Aerospace Conference, NY.
[21] Saha B, Goebel K, Poll S, Christophersen J. Prognostics methods for battery health monitoring using a Bayesian framework. IEEE Transaction on Instrumentation and Measurement 2009;58(2):291–6.
[22] Gao J, Fan W and Han J., On the power of ensemble: Supervised and unsupervised methods reconciled, Tutorial on SIAM Data Mining Conference (SDM), Columbus, OH, 2010.
[23] Breiman L. Bagging redictors. Machine Learning 1996;24:123–40.
[24] Breiman L. Random forests. Machine Learning 2001;45:5–32.
[25] Schapire RE. The strength of weak learnability. Machine Learning 1990;5: 197–227.
[26] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences 1997;55:119–39.
[27] Friedman JH, Popescu BC. Predictive learning via rule ensembles. The Annals of Applied Statistics 2008;2(3):916–54.
[28] Perrone MP, Cooper LN. When networks disagree: ensemble methods for hybrid neural networks. In: Mammone RJ, editor. Neural Networks for Speech and Image Processing. Chapman-Hall; 1993.
[29] Bishop CM. Neural Networks for Pattern Recognition. Oxford University Press; 2005.
[30] Zerpa LE, Queipo NV, Pintos S, Zerpa Salager JL. An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates. Journal of Petroleum Science and Engineering 2005;47(3–4):197–208.
[31] Goel T, et al. Ensemble of surrogates. Structural and Multidisciplinary Optimization 2007;33(3):199–216.
[32] Acar E, Rais-Rohani M. Ensemble of metamodels with optimized weight factors. Structural and Multidisciplinary Optimization 2009;37(3):279–94.
[33] Hu J, Yang YD, Kihara D. EMD: an ensemble algorithm for discovering regulatory motifs in DNA sequences. BMC bioinformatics 2006;7:342.
[34] Chen S, Wang W, Zuylen H. Construct support vector machine ensemble to detect traffic incident. Expert Systems with Applications 2009;36(8): 10976–86.
[35] Baraldi P, Razavi-Far R, Zio E. Classifier-ensemble incremental-learning procedure for nuclear transient identification at different operational conditions. Reliability Engineering and System Safety 2011;98(4):480–8.
[36] Evensen G. The ensemble Kalman filter: theoretical formulation and practical implementation. Ocean Dynamics 2003;53(4):343–67.
[37] Niu G, Yang B, Pecht M. Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance. Reliability Engineering and System Safety 2010;95(7):786–96.
[38] Kohavi R., 1995, A study of cross-validation and bootstrap for accuracy estimation and model selection, Proceedings of the International Joint Conference on Artificial Intelligence – IJCAI'95'.
[39] Saxena A, and Goebel K., 2008, Damage propagation modeling for aircraft engine run-to-failure simulation, IEEE International Conference on Prognostics and Health Management, Denver, CO, Oct 6–9.
[40] Lee WR, Jung SW, Yang KH and Lee JS., 2005, A study on the determination of subjective vibration velocity ratings of main transformers under operation in nuclear power plants, 12th International Congress on Sound and Vibration, Paper 1017, Portugal, July 11–14.
[41] Wang M, Vandermaar AJ, Srivastava KD. Review of condition assessment of power transformers in service. IEEE Electrical Insulation Magazine 2002;18(6):12–25.
[42] Ji S, Luo Y, Li Y. Research on extraction technique of transformer core fundamental frequency vibration based on OLCM. IEEE Transactions on Power Delivery 2006;21(4):1981–8.
[43] García B, Member, Burgos JC, Alonso ÁM. Transformer tank vibration modeling as a method of detecting winding deformations—Part II: Experimental verification. IEEE Transactions on Power Delivery 2006;21(1): 164–9.
[44] Xu H, Rahman S. Decomposition methods for structural reliability analysis. Probabilistic Engineering Mechanics 2005;20(3):239–50.
[45] Gebraeel NZ, Lawley MA, Li R, Ryan JK. Residual-life distributions from component degradation signals: a Bayesian approach. IIE Transactions on Reliability 2005;37(6):543–57.
[46] Kwon D, Azarian M, and Pecht M., 2008, Detection of solder joint degradation using RF impedance analysis, IEEE Electronic Components and Technology Conference, Lake Buena Vista, FL, 27–30 May, 606–610.
[47] Smola AJ, Schölkopf B. A tutorial on support vector regression. Statistics and Computing 2004;14(3):199–222.

[48] Canu S, Mary X, Rakotomamonjy A. Functional learning through kernel. Advances in Learning Theory: Methods, Models and Applications 2003;190: 89–110.
[49] Tipping ME. Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research 2001;1:211–44.
[50] 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 Networks 2007;20(2):236–44.
[51] Tian X., 2006, Cooling fan reliability, failure criteria, accelerated life testing, modeling, and quantification, IEEE Annual Reliability and Maintainability Symposium, Newport Beach, CA, Jan 23–26.
[52] Burger R. Cooling tower technology – maintenance, updating and rebuilding. Fairmont Press. 1995.
[53] Lindely DV, Smith AF. Bayes estimates for linear models. Journal of the Royal Statistical Society (B) 1972;34(1):1–41.