Mechanical Systems and Signal Processing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data

Chao Hu [a,1], Byeng D. Youn [b,*], Taejin Kim [b], Pingfeng Wang [c]

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

ARTICLE INFO

Article history:
Received 2 September 2014
Received in revised form
29 December 2014
Accepted 5 March 2015

Keywords:
Co-training
Semi-supervised learning
Suspension data
Data-driven prognostics
RUL prediction

ABSTRACT

Traditional data-driven prognostics often requires some amount of failure data for the offline training in order to achieve good accuracy for the online prediction. Failure data refer to condition monitoring data collected from the very beginning of an engineered system's lifetime till the occurrence of its failure. However, in many engineered systems, failure data are fairly expensive and time-consuming to obtain while suspension data are readily available. Suspension data refer to condition monitoring data acquired from the very beginning of an engineered system's lifetime till planned inspection or maintenance when the system is taken out of service. In such cases, it becomes essentially critical to utilize suspension data which may carry rich information regarding the degradation trend and help achieve more accurate remaining useful life (RUL) prediction. To this end, this paper proposes a co-training-based data-driven prognostic approach, denoted by COPROG, which uses two data-driven algorithms with each predicting RULs of suspension units for the other. After a suspension unit is chosen and its RUL is predicted by an individual algorithm, it becomes a virtual failure unit that is added to the training data set of the other individual algorithm. Results obtained from two case studies suggest that COPROG gives more accurate RUL prediction, as compared to any individual algorithm with no use of suspension data, and that COPROG can effectively exploit suspension data to improve the prognostic accuracy.

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 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 [1–5]. Currently, there are mainly three paradigms for health prognostics, that is, model-based approaches [6–11], data-driven approaches [12–20] and hybrid approaches [21–23]. The application of

* Corresponding author. Tel.: +82 2 880 1919; fax: +82 2 883 1315.
E-mail addresses: huchaostu@gmail.com (C. Hu), bdyoun@snu.ac.kr (B.D. Youn), godori16@snu.ac.kr (T. Kim), pingfeng.wang@wichita.edu (P. Wang).
[1] Current address: Medtronic Energy and Component Center, Medtronic Inc., Minneapolis, MN 55430, USA

http://dx.doi.org/10.1016/j.ymssp.2015.03.004
0888-3270/

general model-based prognostic approaches relies on the understanding of system physics-of-failure and underlying system degradation models. Myötyri et al. [6] proposed the use of a stochastic filtering technique for real-time remaining useful life (RUL) prediction in case of fatigue crack growth while 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 [7]. Luo et al. [8] 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 et al. presented a degradation modeling framework for RUL predictions of rolling element bearings under time-varying operational conditions [9] or in the absence of prior degradation information [10]. Si et al. presented a drift coefficient model for a nonlinear Wiener degradation process and employed a recursive filter algorithm to derive an approximate RUL distribution [11]. As complex 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. Two good reviews of data-driven prognostic approaches were given in [12] and [13]. Data-driven prognostic approaches generally require the sensory data fusion and feature extraction, statistical pattern recognition, and, for the life prediction, the interpolation [13–16], extrapolation [17], and machine learning [18–20]. Hybrid approaches leverage the strengths of model-based and data-driven approaches by fusing the information from both approaches. Kozlowski et al. [21] 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. [22] employed a Dempster–Shafer regression to fuse a physics-based model and an experience-based model for prognostics. Saha et al. [23] combined an offline relevance vector machine with an 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.

In the context of machine learning, traditional data-driven prognostic approaches mentioned in the literature survey above belong to the category of supervised learning which relies on some amount of failure data for the offline training in order to achieve good accuracy in the online prediction. Here, failure data refer to condition monitoring data collected from the very beginning of an engineered system's lifetime till the occurrence of its failure. Unfortunately, in many engineered systems, only very limited failure data are available since running systems to failure can be a fairly expensive and lengthy process. In contrast, we often can easily obtain a large amount of suspension data. Suspension data refer to condition monitoring data acquired from the very beginning of an engineered system's lifetime till planned inspection or maintenance when the system is taken out of service. The lack of failure data and the large amount of suspension data carrying rich information on the degradation trend make it essentially critical to exploit suspension data in order to improve the accuracy in RUL prediction. However, the utilization of both failure and suspension data for data-driven prognostics, which can be treated as semi-supervised learning in the context of machine learning, is still in infancy. The very few relevant works we are aware of are the survival probability-based approaches [24–26] and life-percentage-based approach [7]. The former approaches use condition monitoring data as inputs to an artificial neural network (ANN) [24] or relevance vector machine [25,26] which then produces the survival probability as the output. As pointed out in [27], the drawback of these approaches lies in the fact that the outputs cannot easily be converted to equivalent RULs for practical use. The latter approach employs condition monitoring data and an age value as inputs to an ANN which then produces the life percentage as the output. Although this approach is capable of enhancing the accuracy in RUL prediction, it still suffers from the follows drawbacks: (i) it simply uses all suspension data regardless of the quality and usefulness; and (ii) the only criteria to determine the RUL of a suspension unit is the minimization of a validation error in the offline training, which could lead to a largely incorrect RUL estimate or even a physically unreasonable estimate (i.e., less than or equal to zero) of that unit.

Recently, the co-training regression has been recognized as one of the main paradigms of semi-supervised learning [28–30], but its usefulness in data-driven prognostics has not been investigated. In this paper, a co-training-based data-driven prognostic approach, named COPROG (i.e., Co-training prognostics), is proposed as the first attempt to derive a semi-supervised learning framework for data-driven prognostics. This approach employs two individual data-driven algorithms, each of which predicts the RULs of suspension units iteratively for the other during the training process. After the RUL of a suspension unit is predicted by an individual algorithm, it becomes a virtual failure unit that is added to the training data set. In order to choose the appropriate suspension unit to use, COPROG quantifies the confidence of an algorithm in predicting the RUL of a suspension unit by how much the inclusion of that unit in the training data set reduces the sum of squared errors (SSE) in RUL prediction on the training data set. The process of iterative training is repeated until there is no suspension unit that is capable of reducing the SSE of any individual algorithm on the training data set or the maximum number of co-training iterations is reached. The final RUL prediction is performed by combining the RUL estimates produced by both individual algorithms. The underlying idea of COPROG is to facilitate an effective exploitation of the degradation trend information carried by suspension data in order to improve the generalization ability of each individual algorithm and achieve more accurate RUL prediction.

The remainder of this paper is organized as follows. Section 2 gives a brief introduction to the two data-driven prognostic algorithms, the feed-forward neural network (FFNN) and the radial basis network (RBN), that are used in this study. Section 3 presents the proposed co-training approach. Applications of the proposed approach are presented in Section 4. The paper is concluded in Section 5.

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

2. Data-driven prognostic algorithms

An artificial neural network (ANN) can be treated as a non-linear model that establishes a set of interconnected functional relationships between input patterns and desired outputs where a training process is employed to adjust the parameters (mainly network weights) of the functional relationships to achieve optimal performance. In recent years, neural networks have been extensively applied to predicting RULs in various contexts such as machinery prognostics [24,27], flight control prognostics [31,32] and battery prognostics [33]. In the context of prognostics, a regression algorithm takes the (processed) sensory signals as the inputs and produces the RUL prediction as the output and can be treated as a data-driven prognostic algorithm. This section briefly introduces two selected neural network approaches for data-driven prognostics: the FFNN and the RBN. A validation mechanism with multiple trials is used to train both the FFNN and RBN with an aim to minimize overfitting as well as to improve generalization.

2.1. Feed-forward neural network

2.1.1. Network structure

The feed-forward neural network (FFNN), also known as the simplest type of ANN, can fit any finite input–output mapping problem with a sufficient number of neurons in the hidden layer [35]. The network is typically composed of three layers (see Fig. 1), namely, the input layer I, hidden layer H, and output layer O. The numbers of the input, hidden and output units are denoted as $|I|$, $|R|$ and $|O|$, respectively. Units of the input layer and the hidden layer are fully connected through the weights $\mathbf{W}^{IH}$ while units of the hidden layer and output layer are fully connected through the weights $\mathbf{W}^{HO}$. The activation functions used in the hidden and output layers are, respectively, the hyperbolic tangent sigmoid transfer function and the linear transfer function.

For data-driven prognostics, the inputs to the FFNN are the normalized age value of a system unit at the current measurement point and the normalized sensory measurements at the current and previous measurement points. If we have N_s sensory measurements (vibration, temperature, voltage, etc.) as the condition monitoring data at each measurement point, the vector $\mathbf{I}^{(t)}$ of network input patterns at a time instance t is denoted by an input vector $\mathbf{x}=(x_1, x_2,\ldots, x_{2Ns+1})$ with x_1 being the normalized age value at the current measurement point, x_{2i} and x_{2i+1} being the ith sensory measurement at the current and previous measurement points, respectively, for $1 \le i \le N_s$. The output is the predicted normalized RUL at the current measurement point, denoted by L^P. As pointed out in previous works [27,34], the combined use of two consecutive measurement sets provides valuable information regarding the rate of change of sensory measurements and thus the rate of system health degradation. We intend not to use more than two data points due to the following reasons: (i) more out-of-date information regarding the "trend" of sensory measurements is carried by earlier data points, the addition of which may lead to the distortion of the most up-to-date information obtained from the two most recent data points; and (ii) an increase in the number of input patterns causes an increase in the network weights to be trained, which results in a higher chance of overfitting and deteriorating the generalization performance. It is noted, though, that the two data points method (using two consecutive data points) employed in this study may be more prone to overfitting than a moving average method using more than two consecutive data points. As the focus of this paper is on the development and verification of a co-training prognostic approach, we intend not to investigate whether the addition of a signal preprocessing step using moving average would improve the prognostic performance of the two prognostic algorithms (FFNN and RBN), even though such an investigation may be of value to the prognostic community. In order to use the FFNN for RUL prediction, the network weights need to be determined through the network training which will be detailed in the subsequent section.

2.1.2. Training process

The training of FFNN refers to the adjustment of network parameters (weights) by exposing the network to a set of training input instances, observing the network outputs, and readjusting the parameters to minimize a training error. With the improvement of generalization being the main focus of FFNN training, we employ a validation mechanism where the original training data set is divided into two mutually exclusive subsets called a training set and a validation set. The training set is used for computing the gradient and updating the network weights with an aim to improve the prediction accuracy on

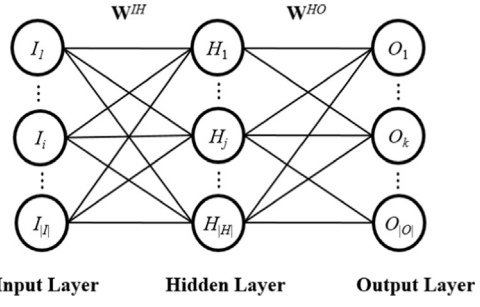

Fig. 1. Structure of a FFNN with one hidden layer.

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

the validation set. The prediction accuracy is quantified using the SSE performance function (or the validation error), expressed as

$$SSE = \sum_{k=1}^{N} e_k^2 = \sum_{k=1}^{N} \left(L_k^P - L_k^T\right)^2 \tag{1}$$

where N is the number of input and output instances in the validation set, e_k is the prediction error for the kth instance, and L_k^P and L_k^T are the predicted and true normalized RULs for the kth instance. The validation error typically decreases during the first few training iterations, and as overfitting starts to occur, the error normally starts to increase. The training is stopped when the increase in the validation error lasts for a specified number of training iterations. The network weights at the minimum validation error are used for RUL prediction.

2.2. Radial basis network

2.2.1. Network structure

Another ANN approach we employ for data-driven prognostics is the radial basis network (RBN) which was reported to have important universal approximation properties [36], and whose structure bears a striking resemblance to that of FFNN in Fig. 1. In an RBN, each unit in the hidden layer is a radial basis function ϕ with its own center, and for each input pattern $\mathbf{x}=(x_1, x_2,\ldots, x_{2Ns+1})$, it computes the Euclidean distance between $\mathbf{x}$ and its center and then applies a polyharmonic basis function, expressed as

$$\phi(\mathbf{x}, \mathbf{c}_j) = \begin{cases} \|\mathbf{x}-\mathbf{c}_j\|^{k_j}, & k_j = 1, 3, 5, \ldots \\ \|\mathbf{x}-\mathbf{c}_j\|^{k_j} \ln(\|\mathbf{x}-\mathbf{c}_j\|), & k_j = 2, 4, 6, \ldots \end{cases} \tag{2}$$

where $\mathbf{c}_j$ and k_j are the center and function order of the jth unit in the hidden layer. From the above expression, it can be observed that each hidden unit in the RBN computes an output that depends on a radially symmetric function. As the input moves closer to the center of the hidden unit, the output becomes smaller, and when the input overlaps with the center of the hidden unit, the smallest output (i.e., zero) can be obtained. The network output is the predicted normalized RUL L^P, expressed as a weighted summation of the outputs of hidden units

$$L^P = \sum_{j=1}^{M} W_{kj}^{HO} \phi(\mathbf{x}, \mathbf{c}_j) \tag{3}$$

In order to use the RBN for RUL prediction, both the centers of hidden units and network weights need to be determined through the network training which will be detailed in the subsequent section.

2.2.2. Training process

The training of an RBN can be viewed as a curve-fitting problem in a multidimensional space from the following two perspectives: (i) the objective of the training is to find an optimal response surface in a multidimensional space that provides the best fit to the training instances; and (ii) the testing (i.e., the test input data are not seen before) is equivalent to the use of this multidimensional surface to interpolate the test data. In this study, a two-phase learning scheme [37] is used to train the RBN with the multivariate polyharmonic basis function as the activation function. This training process is detailed as follows:

Phase 1: Initialize the centers $\mathbf{C}$ of radial basis functions (RBFs) from training input instances randomly selected from the original training data set, $\mathbf{C}=[\mathbf{c}_1,\ldots, \mathbf{c}_M]$ with $\mathbf{c}_j$ being the jth RBF center. The RBF center c_j is of the same format as the network input pattern $\mathbf{x}$, i.e., $c_j=(c_{1j}, c_{2j},\ldots, c_{(2Ns+1)\ j})$ with c_{1j} being the current age value of the selected training input instance, and x_{2i} and x_{2i+1} being the $(i-1)$th sensory measurement of the selected training input instance at the current and previous measurement points, respectively, for $1 \le i \le N_s$. We note that, besides the random selection of training input instances, the RBF centers can also be determined by unsupervised clustering or supervised vector quantization [37].

Phase 2: Determine the output layer weights $\mathbf{W}^{HO}$ which best approximate the training instances by a matrix pseudo-inverse technique, expressed as

$$\mathbf{W}^{HO} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}} \tag{4}$$

where the target output vector $\mathbf{L}^{\mathrm{T}}=[\mathbf{L}_1^{\mathrm{T}},\ldots, \mathbf{L}_N^{\mathrm{T}}]^{\mathrm{T}}$, and $\mathbf{\Phi}$ is an $N\times(M+1)$ design matrix constructed based on the training instances and RBF centers with $\Phi_{ij}=\phi(\mathbf{x}_i,c_j)$. The gradient-descent error backpropagation learning method is not used in this study because, compared to the matrix pseudo-inverse technique, it requires much higher computational effort.

3. Co-training prognostics

This section presents the proposed co-training approach for data-driven prognostics. Section 3.1 describes the overall procedure of this approach. Section 3.2 details the measure to quantify the confidence of an individual data-driven algorithm in predicting the RUL of a suspension unit. Section 3.3 is dedicated to introducing the weight optimization scheme

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

for combining RUL estimates from two algorithms for online prediction. Remarks on how COPROG can help improve the prognostic performance are given in Section 3.4.

3.1. Overall procedure

In the context of machine learning, the two data-driven prognostic algorithms (FFNN and RBN) detailed in Section 2 can be treated interchangeably as two regressors whose focus is to model the relationship between the RUL (dependent variable) and the current age value and sensory measurements (independent variables). Similarly, failure data can be treated as labeled data since each input instance (age value and sensory measurements) has its corresponding label (RUL), while suspension data can be named as unlabeled data since the RUL (label) of each input instance is unknown. Then, the process of estimating the RUL (label) for a suspension unit (unlabeled data) can be treated as the labeling process.

Let $\mathscr{L}=\{(\mathbf{x}_1,L_1^T),\ldots,(\mathbf{x}_{|\mathscr{L}|},L_{|\mathscr{L}|}^T)\}$ and $\mathscr{U}$ represent the failure (labeled) and suspension (unlabeled) training data sets, respectively, where $\mathbf{x}_i$ is the ith input instance composed of $2N_s+1$ elements, L_i^T is its normalized RUL (label), $|\mathscr{L}|$ is the number of labeled instances, and the RULs (labels) of instances in $\mathscr{U}$ are unknown. The pseudo-code of COPROG is shown in Table 1, where the function *TrainFun*($\mathscr{L}$, j) returns the jth trained algorithm ($j=1$ for FFNN and $j=2$ for RBN) based on the labeled data set $\mathscr{L}$. Inspired by the co-training regression work [28–30], the training process in COPROG (see Fig. 2) is designed to work as follows: initially, two algorithms h_1 and h_2 are trained based on the failure data set $\mathscr{L}$ (line 2 of the pseudo-code) and, during the subsequent iterations, the refinement of each algorithm is executed with the help of unlabeled instances (from suspension units) labeled by the other algorithm (lines 4–21 of the pseudo-code). During each

Table 1
Pseudo-code of COPROG

Input: $\mathscr{L}$ – failure training data set, $\mathscr{U}$ – suspension training data set, T – maximum number of co-training iterations, u – suspension pool size
Training Process:
1 $\mathscr{L}_1=\mathscr{L}$; $\mathscr{L}_2=\mathscr{L}$
2 $h_1=TrainFun(\mathscr{L}_1, 1)$; $h_2=TrainFun(\mathscr{L}_2, 2)$;
3 **Repeat** for T times
4 Create a pool $\mathscr{U}'$ of u suspension units by random sampling from $\mathscr{U}$
5 **for** $j=1$ to 2
6 **for** each $X_u \subset \mathscr{U}'$
7 $\mathbf{L}_u^P=h_j(\mathbf{X}_u)$;
8 $h_j'=TrainFun(\mathscr{L}_j\cup\{\mathbf{X}_u, \mathbf{L}_u^P\}, j)$;
9 $\Delta_{j,\mathbf{x}_u}=\sum(L_i^T - h_j(\mathbf{x}_i))^2 - \sum(L_i^T - h_j'(\mathbf{x}_i))^2$
10 **end**
11 **if** there exists an $\Delta_{j,\mathbf{x}_u}>0$
12 $\mathbf{X}_j^*=\arg\max_{Xu\subset\mathscr{U}'}(\Delta_{j,\mathbf{x}_u})$; $\mathbf{L}_j^*=h_j(\mathbf{X}_j^*)$;
13 $\pi_j=\{(\mathbf{X}_j^*, \mathbf{L}_j^*)\}$; $\mathscr{U}'=\mathscr{U}'\backslash\pi_j$;
14 **else**
15 $\pi_j=\emptyset$;
16 **end**
17 **end**
18 **if** $\pi_1==\emptyset$ && $\pi_2==\emptyset$ **exit**
19 **else** $\mathscr{L}_1=\mathscr{L}_1\cup\pi_2$; $\mathscr{L}_2=\mathscr{L}_2\cup\pi_1$;
20 $h_1=TrainFun(\mathscr{L}_1, 1)$; $h_2=TrainFun(\mathscr{L}_2, 2)$;
21 **end**
Testing Process:
22 $L^P=w_1h_1(\mathbf{x})+w_2h_2(\mathbf{x})$ for any test data $\mathbf{x}$

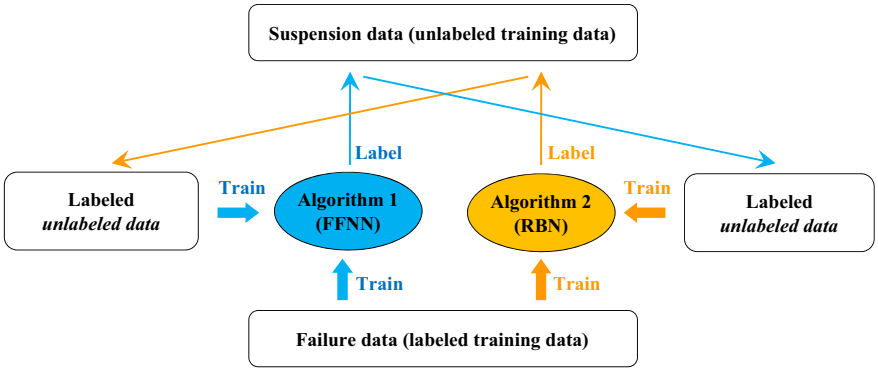

Fig. 2. Flowchart of training process in COPROG.

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

iteration, a set $\mathcal{U}'$ of u suspension units is randomly sampled from $\mathcal{U}$ (line 4 of the pseudo-code). Each trained algorithm h_j then predicts the RULs (labels) of input instances from each suspension unit in $\mathcal{U}'$ (lines 6–10 of the pseudo-code) and selects the unit $\mathbf{X}_j^*$ with the highest labeling confidence (lines 11–16 of the pseudo-code). The selection of suspension units will be discussed in the subsequent section. After an RUL estimate produced by a trained algorithm is assigned to a suspension unit, all the unlabeled instances (with unknown RULs) from this suspension unit will have their RULs (labels) identified and thus become the so-called *labeled* unlabeled instances. The other algorithm is then refined with the *labeled* unlabeled instances $\pi_j=\{(\mathbf{X}_j^*, \mathbf{L}_j^*)\}$ added to its training data set $\mathcal{L}_j$ (lines 18–21 of the pseudo-code). The process of iterative training is repeated until the predefined maximum number T of iterations is reached (line 3 of the pseudo-code) or no suspension unit can be found to be capable of reducing the prediction error of either algorithm on its training data set (line 18 of the pseudo-code). Note that a failure or suspension unit contains multiple input instances and, to distinguish a failure/suspension unit from an input instance, we use the uppercase notation $\mathbf{X}$ to denote the former and the lowercase notation $\mathbf{x}$ to denote the latter. In the testing process, the RUL estimate for an unlabeled input instance is the weighted sum of the RUL estimates of the input instance by the two individual algorithms that are built after the last COPROG iteration (line 22 of the pseudo-code).

3.2. Confidence measure

An inappropriate selection of a suspension unit may lead to incorrect RUL estimation (or mislabeling) on that unit by a prognostic algorithm. The mislabeled suspension unit, if added to the training data set, may negatively affect the performance of the algorithm. Therefore, it is important to choose an appropriate suspension unit to utilize in each co-training iteration. In order to identify the appropriate suspension unit, the confidence in labeling a suspension unit should first be evaluated for all the suspension units and then the one with the highest labeling confidence should be chosen to be utilized. Intuitively, the most confidently labeled suspension unit by a prognostic algorithm, if utilized by the algorithm, should reduce the error of the algorithm to the greatest extent. Thus, the confidence in labeling a suspension unit can be quantified by the extent to which the inclusion of that unit in the training data set reduces the SSE in RUL prediction on the training data set. Mathematically, the confidence measure of the jth algorithm on a suspension unit $\mathbf{X}_u$ can be expressed as

$$\Delta_{j,\mathbf{X}_u} = \sum_{\mathbf{x}_i \in \mathcal{L}_j} \left(\Delta_{\mathbf{x}_i} - \Delta'_{\mathbf{x}_i}\right) \tag{5}$$

The first term, $\Delta_{\mathbf{x}_i}$, in Eq. (5) represents the RUL prediction error for the training input instance $\mathbf{x}_i$ by the jth prognostic algorithm before the inclusion of $\mathbf{X}_u$, calculated as

$$\begin{aligned}\Delta_{\mathbf{x}_i} &= \left(L_i^T - L_j^P(\mathbf{x}_i, \mathcal{L}_j)\right)^2 \\ &= \left(L_i^T - h_j(\mathbf{x}_i)\right)^2\end{aligned} \tag{6}$$

where L_i^T denotes the true RUL of the input instance $\mathbf{x}_i$ in the labeled training data set $\mathcal{L}_j$, and $L_j^P(\mathbf{x}_i, \mathcal{L}_j)$ denotes the predicted RUL of the input instance $\mathbf{x}_i$ by the jth prognostic algorithm trained with the labeled training data set $\mathcal{L}_j$ (or the original jth prognostic algorithm h_j). The second term, $\Delta'_{\mathbf{x}_i}$, in Eq. (5) denotes the RUL prediction error by the jth prognostic algorithm after the inclusion of $\mathbf{X}_u$, together with the predicted RULs L_u^P of its input instances, in the training data set, expressed as

$$\Delta'_{\mathbf{x}_i} = \left(L_i^T - L_j^P\left(\mathbf{x}_i, \mathcal{L}_j \cup \left\{\mathbf{X}_u, \mathbf{L}_u^P\right\}\right)\right)^2 = \left(L_i^T - h'_j(\mathbf{x}_i)\right)^2 \tag{7}$$

where L_u^P denotes the predicted RULs of the input instances in the suspension unit $\mathbf{X}_u$ and $L_u^P = h_j(\mathbf{X}_u)$, and $h_{j'}$ denotes the RUL predicted by the refined jth prognostic algorithm that has utilized the information provided by the suspension unit $\{\mathbf{X}_u, L_u^P\}$. With the defined confidence measure, the suspension unit with the highest labeling confidence can be selected by

$$\mathbf{X}_j^* = \underset{\mathbf{X}_u \in U'}{arg\,max}(\Delta_{j,\mathbf{X}_u}) \tag{8}$$

The selected suspension unit, together with the predicted RULs of its input instances, can then be included in the labeled training data set to improve the accuracy of the jth prognostic algorithm. The above confidence measure reflects the fact that the most confidently labeled suspension unit makes the prognostic algorithm most consistent with its existing training data set.

3.3. Weight optimization

After using two data-driven prognostic algorithms to select and label the unlabeled suspension units during the offline training, we then obtain two augmented labeled training data sets $\mathcal{L}_1$ and $\mathcal{L}_2$, each of which contributes a trained algorithm for the online prediction. Then, the RUL predictions of these two algorithms are combined in a weighted-sum formulation as the final prediction. The simplest way to do so is to average the two predictions, which is ideal only when the prognostic algorithms provide the same level of accuracy. However, it is more likely that an algorithm tends to be more accurate than the other. In such cases, a greater weight should be assigned to a member algorithm with higher prediction accuracy so that

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

the algorithm makes more contribution to the ensemble prediction. Hence, two individual algorithms with different prediction performance should be multiplied by different weight factors. In what follows, a weight optimization scheme is employed to maximize the accuracy in RUL prediction by adaptively synthesizing the prediction accuracy of each individual algorithm [16]. In this scheme, the optimum weights can be obtained by solving an optimization problem of the following form

$$\begin{aligned} &\text{Minimize} \quad SSE = \sum_{\mathbf{x}_i \in \mathscr{L}} \left(L_i^T - (w_1 h_1(\mathbf{x}_i) + w_2 h_1(\mathbf{x}_i)) \right)^2 \\ &\text{Subjectto} \quad w_1 + w_2 = 1, 0 \le w_1 \le 1, 0 \le w_2 \le 1 \end{aligned} \tag{9}$$

where $\mathscr{L}$ denotes the labeled training data set. After the prediction of RULs using the two prognostic algorithms, both terms h_1 and h_2 in Eq. (9) become known and, since the weight optimization process does not require the re-execution of these two algorithms, the above optimization problem can be readily solved with almost negligible computational effort. This study employs sequential quadratic programming (SQP) as a numerical optimization method to solve the optimization problem in Eq. (9). We expect that, by solving the optimization problem in Eq. (9), the resulting ensemble of the two algorithms will outperform its counterpart with equal weights in terms of prediction accuracy.

3.4. Rationale for co-training prognostics

In what follows, we intend to elaborate on how COPROG can utilize the suspension data to improve the prognostic performance from two perspectives: (i) how an individual prognostic algorithm can benefit from the utilization of suspension data; and (ii) how the combined use of two algorithms can enhance the prognostic accuracy as compared to an individual algorithm.

3.4.1. Utilization of suspension data

Fig. 3 illustrates, in a prognostic sample space $\mathcal{P}$, that using one prognostic algorithm (FFNN or RBN) to label the unlabeled instances helps improve the prediction accuracy on the testing data. Here, $\mathcal{P}$ consists of all possible prognostic samples obtained under different testing situations (e.g., manufacturing condition, health condition and degradation rate). Sparse labeled data (or failure data) and plenty of unlabeled data (or suspension data) can be used for training. A prognostic algorithm (FFNN or RBN) trained with only the labeled data can generalize sufficiently well to make reasonably accurate predictions on the testing data in the close vicinity of the labeled data. However, not all predictions made by this algorithm are accurate: in regions that are sparsely populated by the labeled data, relatively large errors are expected (as is the case in Fig. 3). In other words, the RUL predictions by the algorithm are expected to contain relatively large errors on the testing data that fall significantly away from the labeled data. If the unlabeled data can be properly labeled (the data then become the *labeled* unlabeled data) and added to the labeled data set, the algorithm will then be trained with an augmented labeled data set and the region where the algorithm can make accurate prediction is expected to be expanded to encompass testing data that are close neighbors of the *labeled* unlabeled data. We note that the *proper labeling* is realized by selecting appropriate unlabeled data through the maximization of the confidence measure in Eq. (5).

3.4.2. Ensemble of prognostic algorithms

The strategy of using two prognostic algorithms produces two unique benefits that are detailed as follows:

Creating diversity: The two algorithms built with different network structures and training procedures lead to the creation of diversity in RUL prediction. Based on the created diversity, the ensemble obtains better predictive performance than could be obtained by any individual algorithm. In addition, the suspension unit chosen by h_1 during each iteration will not be chosen by h_2, and vice versa. Thus, the suspension units that the two algorithms label for each other are different, which can be treated as another mechanism for encouraging the diversity.

Reducing overfitting: If the labeled training data set contains noise, the use of two prognostic algorithms can be helpful to reducing overfitting [28]. Let $\mathcal{N}$ denote the set of noisy data in $\mathscr{L}$. For a suspension (unlabeled) unit $\mathbf{X}_u$, either of the two

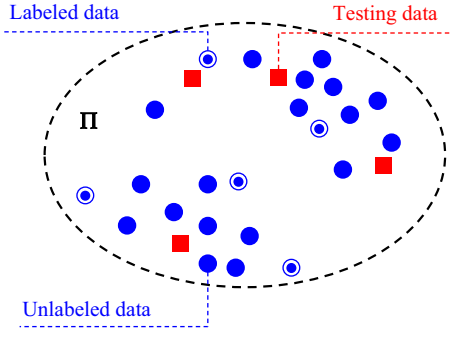

Fig. 3. Prognostic space with labeled, unlabeled and testing data.

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

algorithms h_1 and h_2 labels this unit based on the input–output relationship that is learned through a set of neighboring labeled data of this unit. Here, the neighboring labeled data refer to the labeled training instances whose inputs exhibit high similarity to those of the unlabeled training instances from the suspension unit. Assume the sets of neighboring labeled data for h_1 and h_2 are Ω_1 and Ω_2, respectively, and the two sets are usually different. First, $\mathbf{X}_u$ is labeled by h_1. Then, $\{\mathbf{X}_u, h_1(\mathbf{X}_u)\}$ is added to $\mathscr{L}_1$, where the labels $h_1(\mathbf{X}_u)$ suffers from the noisy data in $\Omega_1 \cap \mathcal{N}$. For another suspension (unlabeled) unit $\mathbf{X}_v$ that we assume is very close to $\mathbf{X}_u$, the neighboring labeled data for labeling $\mathbf{X}_v$ will be approximately $\Omega_1 \cup \{\mathbf{X}_u, h_1(\mathbf{X}_u)\}$. Thus, $h_1(\mathbf{X}_v)$ will be roughly affected by $(\Omega_1 \cap \mathcal{N}) \cup \{\mathbf{X}_u, h_1(\mathbf{X}_u)\}$. Note that $\{\mathbf{X}_u, h_1(\mathbf{X}_u)\}$ has already suffered from the noisy data in $\Omega_1 \cap \mathcal{N}$. Thus, $h_1(\mathbf{X}_v)$ will be affected by $\Omega_1 \cap \mathcal{N}$ more seriously than $h_1(\mathbf{X}_u)$ does. As we label more suspension units, the effect of noise continues to propagate and becomes more severe. Whereas if the unit $\mathbf{X}_u$ is labeled by h_2 and $\{\mathbf{X}_u, h_2(\mathbf{X}_u)\}$ is put into $\mathscr{L}_1$, then $h_1(\mathbf{X}_v)$ will suffer from $\Omega_1 \cap \mathcal{N}$ only once, thereby preventing the effect of noise from propagating.

4. Case studies

In this section, the proposed COPROG approach for data-driven prognostics is demonstrated with two PHM case studies: (i) rolling-element bearing problem (simulation), and (ii) electric cooling fan problem (experiment). To study how the exploitation of suspension data affects the prognostic performance, we compared the prognostic performance of the co-training approach to that of the FFNN and RBN with no use of suspension data.

4.1. Rolling-element bearing problem

The rolling-element bearing is a critical component in rotating machines, since an unexpected failure of the bearing leads to machine shut-down and catastrophic damage. Thus, it is very important to ensure high reliability and safety of the bearing during its operation. This case study conducts bearing health prognostics with sensory signals obtained from a vibration model of the rolling-element bearing.

4.1.1. Bearing defect simulation

We employed an existing vibration model [38,39] to simulate the vibration signal produced by a single point defect on the inner race of a rolling-element bearing under constant radial load. The model takes into account the effects of the single point defect, shaft speed, bearing load distribution, the exponential decay of vibration, and the measurement noise present in a practical measurement system. The simulation assumes the following parameters: pitch angle $\theta = 0^o$, pitch diameter $d_p = 23$ mm, roller diameter $d_r = 8$ mm and number of rollers $n_r = 9$, shaft rotational speed $v_r = 100$ rpm corresponding to shaft rotational frequency $f_r \approx 1.67$ Hz, bearing-induced resonant frequency $f_s = 5000$ Hz, sampling frequency $f_s = 15000$ Hz, and measurement duration $T_s = 2$ s. The characteristic defective frequency corresponding to an inner race fault can be computed as

$$f_{IRF} = \frac{n_r f_r}{2}\left(1 + \frac{d_r}{d_p}\cos(\theta)\right) \approx 10.11\ Hz \tag{10}$$

Fig. 4(a) plots the simulated vibration signal of a bearing with an inner race fault in the time domain. Using the fast Fourier transform (FFT), we converted this signal to the frequency domain and obtained its frequency spectrum in Fig. 4(b) where the spectrum is dominated by high-frequency resonant signals. Through band-pass filtering and rectifying the raw vibration signal, we excluded the resonant signals by other parts of the rotating machine and derived a demodulated signal as shown in Fig. 4(c). The frequency domain plot of the demodulated signal in Fig. 4(d) indicates the presence of a defect with the characteristic frequency of 10.13 Hz which exhibits good consistency with the calculated inner race fault frequency in Eq. (10).

The defect progression was realized by increasing the amplitude of the impulse due to the defect. To model the trajectory of the amplitude of the impulse over time, this study uses a damage propagation model of an exponential form, expressed as [16,40]

$$D(t) = D_0 + b_D(1 - \exp(a_D t)) \tag{11}$$

where D_0 is the initial amplitude of the impulse, a_D and b_D are the model parameters that determine the rates of the defect progression, and t is the time instance (in cycles). This study considers the model parameters D_0, a_D and b_D as independent random variables whose statistical information is summarized in Table 2. The failure criterion is defined as the amplitude of the impulse D being 1. The vibration signals were repeatedly generated based on the defect amplitudes at different time instances that exponentially and randomly increases according to Eq. (11). The data generation process for one bearing unit involves three sequentially executed steps: (i) one set of samples for D_0, a_D and b_D are randomly generated according to the statistical information in Table 2; (ii) the set of randomly generated samples are used in conjunction with Eq. (11) to produce a damage propagation path that is then corrupted with random noise following a zero mean normal distribution; and (iii) the vibration signals are simulated by using the vibration model [38,39] for each time instance along the damage propagation path.

The lifecycle evolution of vibration spectra of an example bearing unit is plotted in Fig. 5(a) where we can observe that, as degradation progresses over time, the defect amplitude at harmonic defective frequencies (positive integer multiples of

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

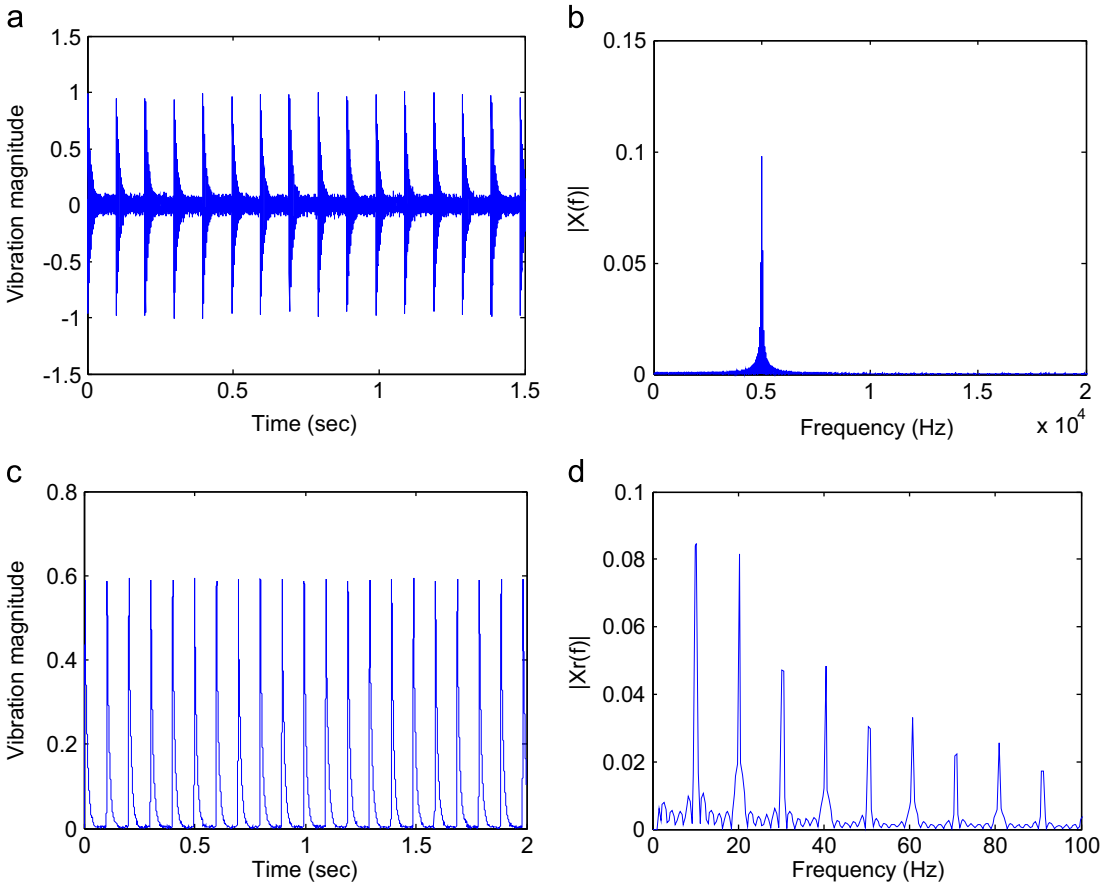

Fig. 4. Simulated signal of outer-race defect: (a) time domain plot and (b) frequency spectrum of raw signal; (c) time domain plot and (d) frequency spectrum of demodulated signal.

Table 2
Statistical information of random variables for defect progression.

Random variable	Distribution type	Mean	Standard deviation	Parameters for non-normal distributions
D_0	Normal	0.010	0.001	–
a_D	Lognormal	0.100	0.020	$\mu=-2.322$, $\sigma=0.198$
b_D	Normal	0.025	0.005	–

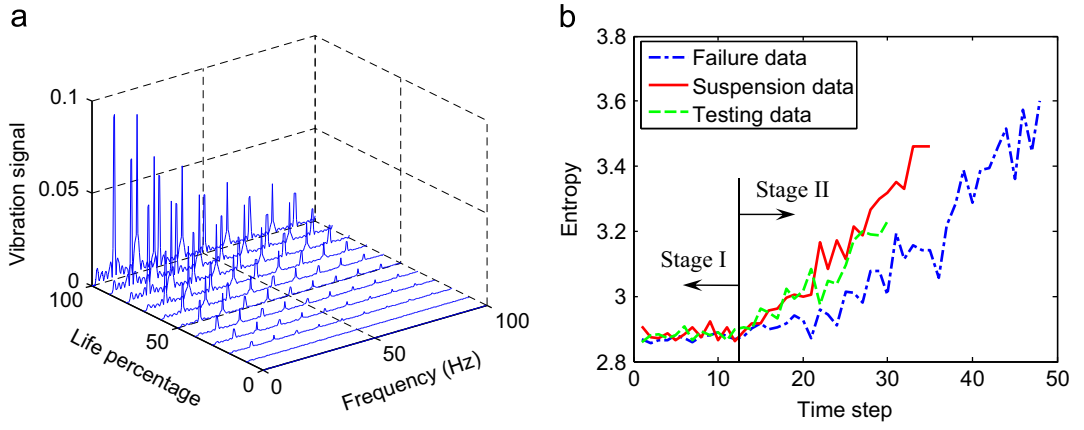

Fig. 5. Lifetime evolution of vibration spectra (a) and entropy (b) with an inner race defect.

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

the characteristic defective frequency) begin to appear and increase exponentially. The feature we employed as the input to the FFNN and RBN for prognostics is the entropy (see Fig. 5(b)) which is a statistical measure of randomness used to quantify the uncertainty of the noisy signal [41]. It can be observed from both figures that the degradation undergoes two distinct stages. The first stage is referred to as normal operation period that is characterized by a relatively flat region. In this stage, no obvious defect can be found in the bearing. In the second stage, the degradation of the bearing begins and the signal is characterized by exponentially increasing defect amplitudes. This two-stage degradation behavior is consistent with previous works on bearing prognostics [9,10,42].

4.1.2. Construction of prognostic data sets

For the training process, we generated a training data set that consists of 100 failure (labeled) units and 100 suspension (unlabeled) units. As shown in Fig. 5(b), the failure data contain complete degradation information while the suspension data carry only partial degradation information. The latter were generated by truncating the original failure data after pre-assigned suspension times. The suspension time pre-assigned to each suspension unit was randomly generated from a uniform distribution between 90 and 100 percentile lives. This range was selected based on the assumption that the suspension unit is taken out of service when it approaches its end of life. For the testing process, we first generated a testing data set consisting of 100 testing (failure) units and then produced multiple testing input instances from each testing unit by truncating the failure trajectory of the unit after pre-assigned RULs. The RULs pre-assigned to each testing unit were a series of successive positive integers in the interval $[1, Z]$ where the integer Z was randomly generated from a uniform distribution between 1 and the unit's half-life. In other words, each testing unit was used to produce multiple testing input instances in the second half of the unit's lifetime. A truncated failure trajectory (testing input instance) of a testing unit is plotted in Fig. 5 (b), where we can observe only a portion of the degradation pathway as opposed to the complete degradation pathway in the failure data.

4.1.3. Implementation of COPROG

To investigate the effect of the amount of failure data on the performance improvement by COPROG, we evaluated algorithms under two different settings: *Setting 1* (lack of failure data) – 3 failure units and 10 suspension units (i.e., 3L-10U) and *Setting 2* (plenty of failure data) – 10 failure units and 10 suspension units (i.e., 10L-10U). To comprehensively test the performance of the algorithms with various training sets of failure and suspension data as well as account for the randomness in the training of FFNN and RBN, we repeatedly executed the training and testing processes 50 times, each with a training set of failure and suspension units with the predefined quantities (3L-10U or 10L-10U), and computed the mean (accuracy) and standard deviation (robustness) of root mean square errors (RMSEs) on the testing data set. During each repetition, the failure and suspension units in the training set were randomly selected from the training data set consisting of 100 failure units (labeled) and 100 suspension units (unlabeled). Mathematically, the mean RMSE can be expressed as

$$\mu_{RMSE} = \frac{1}{50}\sum\nolimits_{1 \le k \le 50} RMSE_k = \frac{1}{50}\sum\nolimits_{1 \le k \le 50} \sqrt{\frac{\sum_{x \in \mathcal{T}} \left(L^T(\mathbf{x}) - L_k^P(\mathbf{x})\right)^2}{N_t}} \tag{12}$$

where $L^T(\mathbf{x})$ denotes the true RUL of the testing input instance $\mathbf{x}$, $L^P(\mathbf{x})$ denotes the predicted RUL by an algorithm, and N_t denotes the number of input instances in the testing data set $\mathcal{T}$. As described in Section 4.1.2, the true RUL of a testing input instance is a pre-assigned integer that was used to produce this input instance during the construction of the testing data set. Since the RUL prediction at a late stage exerts a larger influence on maintenance decision-making than that at an early stage, we intended to separately investigate the prognostic accuracy when a bearing approaches its end of life. For this purpose, we defined a special time period, namely the critical time, as the last 5 time instances of each testing trajectory, extracted 5 testing input instances in this time period for each testing trajectory and computed a critical-time RMSE using Eq. (12).

For the implementation of COPROG, the normalized age value of a bearing unit at the current time instance and the normalized entropy feature at the current and previous time instances were used as the inputs to the FFNN and RBN and the output is the normalized RUL at the current time instance. In the FFNN structure, the numbers of the input, hidden and output units are $|I|=3$, $|R|=8$ and $|O|=1$. The FFNN training used 60% of the original data set as the training data set and the rest as the validation set. The network structure and the ratios of the training and validation sets were empirically determined based on the modeling and generalization performance of the trained FFNN model. It was observed that this setting results in a trained FFNN model with good modeling and generalization performance. The backpropagation training with an adaptive learning rate [35] was employed to obtain the optimal weights of the FFNN. The training epochs were set to 100. Since the training algorithm is random, resulting in slightly different SSE values produced by different training executions, we trained the FFNN 10 times to obtain 10 trained FFNNs among which the one with the lowest SSE was selected as the trained model. In the RBN structure, the numbers of the input and output units are 3 and 1, respectively, and the hidden layer consists of 20 RBF centers with first-order polyharmonic functions. The RBN training employed the matrix pseudo-inverse technique [37] to calculate the weights of the output layer in Eq. (4). With an aim to improve the generalization performance of the RBN, we divided the original training data set into the mutually exclusive training set

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

(60% of the original set) and validation set (40% of the original set), trained the RBN with randomly selected RBF centers and evaluated the validation error of the trained RBF 10 times, and selected the one with the lowest validation error as the trained model. In COPROG, both the maximum number of co-training iterations T and the suspension pool size u were set to 5. It is noted that a large suspension pool size (u) could incur high computational cost required by the co-training process (since a large number of suspension units need to be evaluated in each co-training iteration), while a small size may result in a low chance of the suspension data pool containing at least one usable suspension unit. We found that a pool size of 5 achieves a good balance between the computational cost and the data coverage.

4.1.4. Results of COPROG

Table 3 summarizes the RMSE results of supervised (FFNN, RBN, Ensemble) and semi-supervised (COPROG) learning. Here, FFNN and RBN refer to initial algorithms before utilizing any suspension data and Ensemble refers to the ensemble of the two initial algorithms, FFNN and RBN. In what follows, we intend to interpret the results from the following two perspectives:

Prognostic accuracy: It can be observed from Table 3 that the COPROG algorithm under any setting always outperforms either of the two initial algorithms and their ensemble in terms of the life- and critical-time mean RMSEs, which verifies that COPROG is capable of exploiting the suspension data to improve the prognostic accuracy. Under the setting with the lack of failure data (i.e., 3L-10U), COPROG achieves the life- and critical-time mean RMSEs of 5.2674 and 4.5505 on the testing data set, 16.26% and 15.19% improvements over the best initial algorithm, RBN, whose mean RMSEs are 6.2905 and 5.3654, respectively, and 12.76% and 11.48% improvements over Ensemble whose mean RMSEs are 6.0379 and 5.1405, respectively. The accuracy improvements over the best initial algorithm can be attributed to (i) the effective utilization of valuable degradation information that is only carried by the suspension data and (ii) the creation of diversity in RUL prediction through the ensemble prediction (see the remarks in Section 3.4). The observation that COPROG achieves less accuracy improvements over Ensemble than over the best initial algorithm suggests that the ensemble prediction, in addition to the exploitation of suspension data, also enhances the accuracy in RUL prediction. As expected, the accuracy improvements become less significant when we have more failure data (i.e., 10L-10U). This is due to the fact that a larger amount of failure data captures more information regarding the degradation trend and leads to a reduced amount of information gained by utilizing the suspension data.

Prognostic robustness: In addition to the prognostic accuracy, we also evaluated the algorithms in terms of the prognostic robustness, that is, the extent to which the performance of an algorithm is insensitive to the variation in the training data. Here, the prognostic robustness was quantified using the standard deviation of RMSEs obtained from 50 random sets of training data. As shown in Table 3, COPROG always performs better than the two initial algorithms and Ensemble, and the performance improvements over the two initial algorithms are less significant than those over Ensemble. The results suggest that the utilization of suspension data and the ensemble prediction by COPROG improve the prediction robustness. The improved performance by COPROG can be attributed to (i) the enrichment of degradation information by exploiting the suspension data and (ii) the creation of diversity in RUL prediction by combining the predictions of the two individual algorithms.

To illustrate the accuracy improvements obtained by exploiting suspension data, the RUL predictions by the initial algorithms (that is, FFNN and RBN trained without the utilization of any suspension data) and final algorithms (that is, FFNN and RBN after the co-training process) under the setting of 3L-10U are plotted for the testing data set in Fig. 6. The testing input instances are sorted by the RULs in an ascending order. Note that, since a testing unit was used to produce multiple testing input instances in the second half of the unit's lifetime, the number (>2000) of testing inputs instances is larger than the number (100) of testing units. The relatively large scatter of RUL predictions around the true curve can be attributed to the lack of failure units (only 3 in this case) as well as the large noise in the entropy feature data (see Fig. 5(b)). It can be observed that, compared to the two initial algorithms, the final algorithms yield RUL predictions that are closer to the true values while eliminating many outliers produced by the initial algorithms. Overall speaking, the final RBN produces slightly better prognostic accuracy than the final FFNN. For example, the life-time mean RMSE (i.e., 5.4018 cycles) of the final RBN is slightly smaller than that (i.e., 5.8384 cycles) of the final FFNN.

Table 3
RMSE results of supervised (FFNN, RBN and Ensemble) and semi-supervised (COPROG) learning for rolling-element bearing problem.

Training data	Statistics	Life-time RMSE (cycles)				Critical-time RMSE (cycles)			
		FFNN	RBN	Ensemble	COPROG	FFNN	RBN	Ensemble	COPROG
3L-10U	Mean	6.3119	6.2905	6.0379	**5.2674**	5.5487	5.3654	5.1405	**4.5505**
	Std[a]	1.2980	1.2593	1.0903	**0.4851**	1.5794	1.3378	1.2105	**0.7659**
10L-10U	Mean	5.2051	5.0116	4.9382	**4.7928**	4.5234	4.2165	4.0789	**4.0406**
	Std	0.3501	0.4143	0.3075	**0.2637**	0.6504	0.6291	0.5585	**0.5108**

[a] Standard deviation.

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

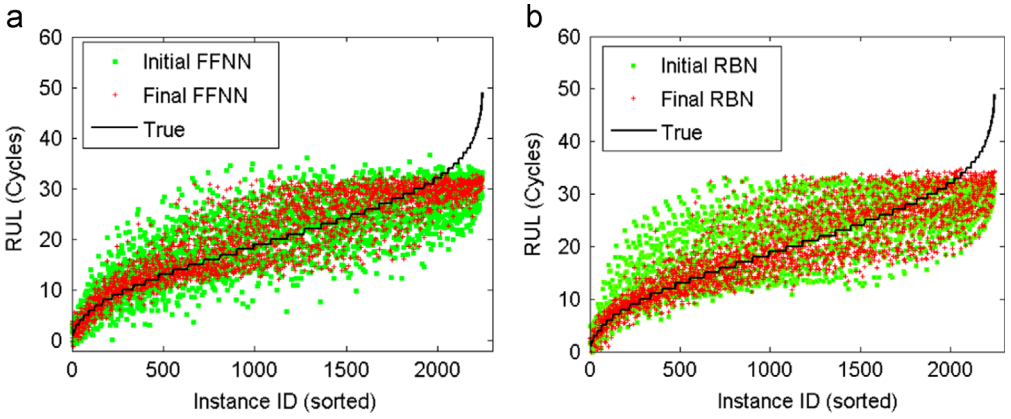

Fig. 6. RUL predictions by initial and final FFNNs (a) and RBNs (b) for rolling-element bearing problem (3L-10U).

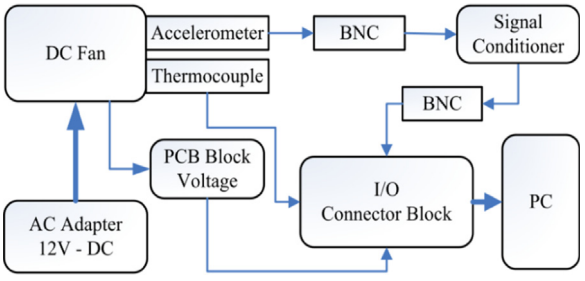

Fig. 7. DC fan degradation test block diagram.

Finally, we note that, in this case study, bearing health prognostics is conducted with the consideration of only one failure cause (i.e., an inner race fault). Multiple nominally identical bearings, if being used in practical applications, could fail due to N_{fc} different failure causes (e.g., inner race fault, outer race fault and ball damage). Although, in such cases, frequency-domain analysis can be used to identify the failure cause for a specific bearing, data-driven prognostics still requires training data (i.e., failure and suspension data) for training the two prognostic algorithms (FFNN and RBN). In fact, it requires N_{fc} training data sets, each of which captures the bearing degradation behavior for a specific failure cause, and co-training can be executed on each of the N_{fc} training data sets, resulting in N_{fc} sets of co-trained prognostic algorithms. Thus, the benefits that co-training provides in the case of only one failure cause still hold in the case of multiple failure causes.

4.2. Electric cooling fan problem

In addition to the simulation studies, we also conducted experimental studies to verify the effectiveness of COPROG. In this case study, we applied COPROG 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 [43] and in cooling towers of many chemical plants [44].

4.2.1. 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 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. 7. 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. 8(a) shows the test fixture with 4 screws at each corner for the DC fan units. As shown in Fig. 8 (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. 9. 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. 9(a). Two wires were connected to the PCB block of the fan to measure the voltage between two fixed points, as shown in Fig. 9(b). As shown in Fig. 9(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

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

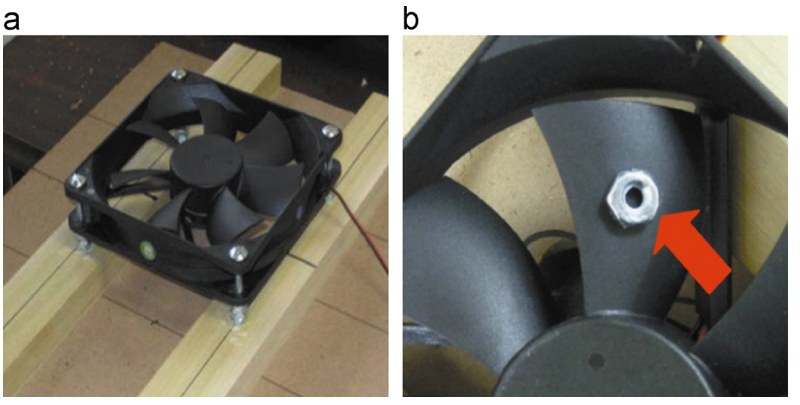

Fig. 8. DC fan test fixture (a) and the unbalance weight installation (b).

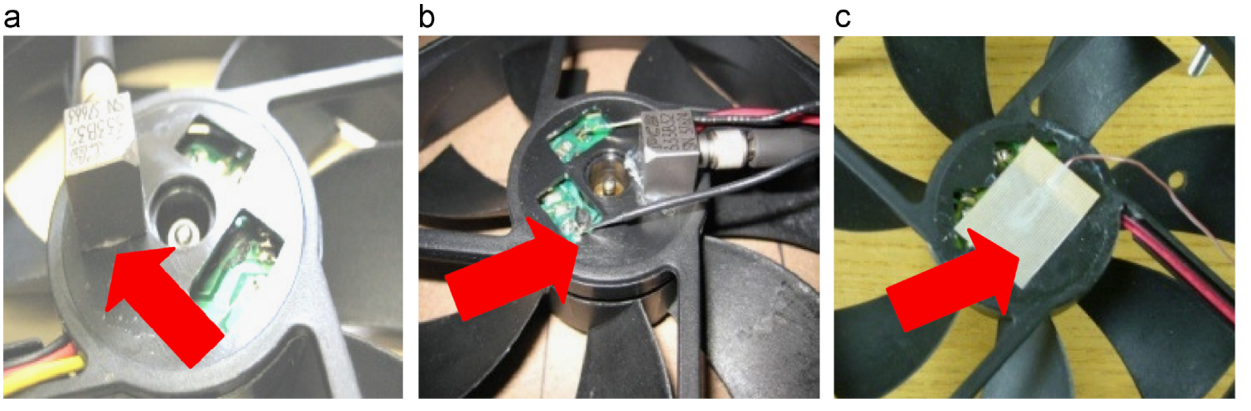

Fig. 9. Sensor installations for DC fan test: (a) accelerometer, (b) voltmeter and (c) thermocouples.

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.

4.2.2. Construction of prognostic data sets

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 input signal to FFNN and RBN for the DC fan prognostics. Fig. 10 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 RMS signal is highly random and non-monotonic because of metal particles, sensory signal noise, and input voltage noise.

Among 32 fan units, 20 fan units were used to construct the training data set consisting of 10 failure (labeled) units and 10 suspension (unlabeled) units, while the rest were used to build the testing data set for the performance evaluation. In this case study, one cycle is defined as every ten minutes. The suspension data were generated by truncating the original failure data after pre-assigned suspension times (in cycles) that were randomly generated from a uniform distribution between 90 and 100 percentile lives (in cycles). The testing data were produced by truncating the failure trajectory of each testing unit after pre-assigned RULs (in cycles). The RULs pre-assigned to each testing unit were a series of successive positive integers between Z and the unit's life (in cycles) where the integer Z was randomly generated from a uniform distribution between 1 and the unit's 20 percentile lives (in cycles).

4.2.3. Implementation of COPROG

The algorithms were evaluated under two different settings: *Setting 1* (lack of failure data) – 3 failure units and 10 suspension units (i.e., 3L-10U) and *Setting 2* (plenty of failure data) – 10 failure units and 10 suspension units (i.e., 10L-10U). We repeatedly executed the evaluation process 20 times under both settings and computed the mean (accuracy) and standard deviation (robustness) of RMSEs on the testing data set. For the first setting, each execution employs a different set of 3 failure units that were randomly selected from the 10 failure units in the training data set. Mathematically, the mean

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

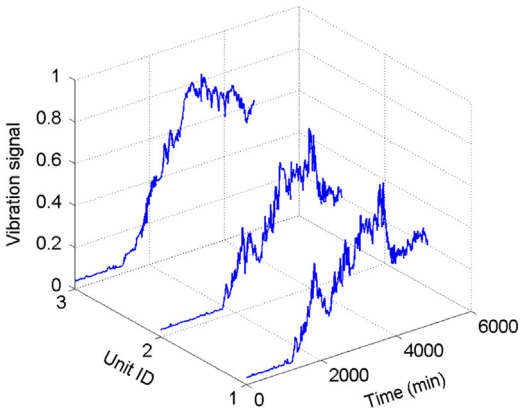

Fig. 10. Sample degradation signals from DC fan testing.

Table 4
RMSE results of supervised (FFNN, RBN and Ensemble) and semi-supervised (COPROG) learning for electric cooling fan problem.

Training data	Statistics	Life-time RMSE (cycles)				Critical-time RMSE (cycles)			
		FFNN	RBN	Ensemble	COPROG	FFNN	RBN	Ensemble	COPROG
3L-10U	Mean	19.0431	19.4701	18.5584	**12.9755**	23.6213	20.5593	19.5372	**13.8701**
	Std	3.4209	4.3246	3.2867	**3.1942**	6.0836	7.1387	4.4735	**2.9440**
10L-10U	Mean	17.1744	16.2880	16.0987	**9.7529**	17.8188	17.1557	16.8067	**10.1301**
	Std	2.3004	2.4111	2.1980	**1.3414**	4.2386	4.1784	3.7390	**1.4219**

RMSE can be expressed as

$$\mu_{RMSE} = \frac{1}{20}\sum_{1 \le k \le 20} RMSE_k = \frac{1}{20}\sum_{1 \le k \le 20} \sqrt{\frac{\sum_{\mathbf{x} \in \mathcal{T}} \left(L^T(x) - L_k^P(x)\right)^2}{N_t}} \tag{13}$$

where $L^T(\mathbf{x})$ denotes the true RUL of the testing input instance $\mathbf{x}$, $L^P(\mathbf{x})$ denotes the predicted RUL by an algorithm, and N_t denotes the number of input instances in the testing data set $\mathcal{T}$. As described in Section 4.2.2, the true RUL of a testing input instance is a pre-assigned integer that was used to produce this input instance during the construction of the testing data set. Since the RUL prediction at a late stage exerts a larger influence on maintenance decision-making than that at an early stage, we intended to separately investigate the prognostic accuracy when a DC fan unit approaches its end of life. For this purpose, we extracted the testing input instances at the last 30 cycles of each testing trajectory and computed a critical-time RMSE using Eq. (13). The parameter settings detailed in Section 4.1.3 were again used for FFNN and RBN training.

4.2.4. Results of COPROG

The RMSE results of the initial algorithms (that is, FFNN and RBN trained before utilizing any suspension data), their ensemble (Ensemble) and COPROG are summarized in Table 4, where we can observe significantly better performance of COPROG than any initial algorithm and their ensemble in terms of both prognostic accuracy and robustness. The results suggest that the exploitation of the suspension data using the co-training approach can help achieve more accurate and stable RUL predictions. It can also be observed that the critical-time RMSEs are larger than the life-time RMSEs under both experimental settings. This counter-intuitive observation can be attributed to the fact that, when a bearing approaches its end of life, the RMS signal exhibits a non-monotonic behavior (see Fig. 10), thereby making accurate RUL prediction more challenging. Under the experimental setting of 3L-10U, the RUL predictions for a sample testing fan unit by COPROG are plotted in Fig. 11 where good accuracy in RUL predictions can be observed.

5. Conclusion

This paper proposed a co-training-based prognostic approach (COPROG), which, to the best of our knowledge, is one of the earliest efforts on semi-supervised learning for data-driven prognostics. By utilizing the suspension data, COPROG achieves better accuracy and robustness in RUL predictions compared to any individual algorithm without utilizing the suspension data. Results from two engineering case studies (rolling element bearing problem and electric cooling fan

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

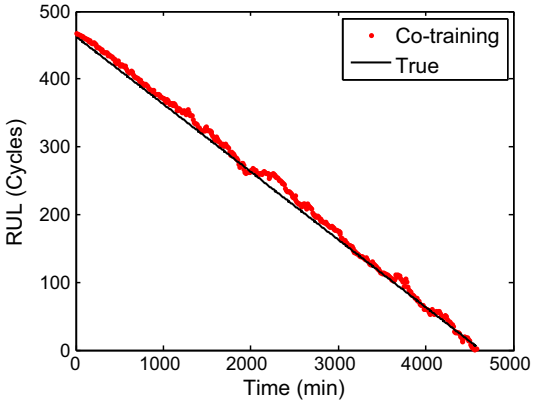

Fig. 11. RUL predictions for a testing fan unit by COPROG (3L-10U).

problem) suggested that COPROG is capable of effectively exploiting the suspension data to improve the prognostic performance and that the improvement becomes more pronounced when we have lack of failure data for the offline training.

In this study, the suspension data are defined as unlabeled condition monitoring data, i.e., the label (RUL) of each input combination in the suspension data is unknown. However, this definition could be generalized beyond the concept of "unlabeled" to the concept of "partially labeled" where the suspension data are supplemented by the health conditions estimated by maintenance personnel (through visual and/or non-visual inspections). It would be interesting to investigate how to utilize the health conditions to improve the performance of semi-supervised prognostics in our future study.

It is noted that the suspension data may be obtained at multiple stages throughout the lifetime of a system unit. This study only considers the suspension data obtained at the last stage of the unit's lifetime, since, intuitively speaking, the degradation trend captured by the suspension data at the last stage most closely resembles the trend during the rest of the lifetime. However, the scope of the suspension data under consideration could be broadened to cover M_s earlier stages of the unit's lifetime, which would effectively expand the suspension training data set by M_s times. It would be interesting to investigate how to appropriately select the additional stages for suspension data expansion.

We also note that a validation approach (e.g., the k-fold cross validation with a training set, and the holdout approach with a training set and an independent validation set), if used to evaluate the accuracy of a given weight combination in the weight optimization, can produce a more representative error measure as compared to the non-validation approach used in this study. Since the weight optimization is not the primary focus of this study, we intend to investigate in our future study how the definition of an error measure affects the performance of the resulting ensemble.

Currently, there are several semi-supervised regression approaches that have recently been developed in the machine learning society. It would be also interesting to investigate the similarity between regression and data-driven prognostics and develop other types of semi-supervised approaches for data-driven prognostics. Furthermore, we observed in our experiments that utilizing unlabeled data does not always help improve performance. Similar phenomena have also been reported by researchers in the machine learning society [28,45]. However, no rigorous guidelines on the exploitation of unlabeled data have yet been established. Future research efforts should be devoted to deriving such guidelines for semi-supervised data-driven prognostics.

Acknowledgments

The work presented in this paper is partially supported by Mid-Career Researcher Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2013R1A2A2A01068627), and the Technology Innovation Program (10050980, System Reliability Improvement and Validation for New Growth Power Industry Equipment) funded by the Ministry of Trade, Industry & Energy (MI, Korea).

References

[1] R. Dekker, Applications of maintenance optimization models: a review and analysis, Reliab. Eng. Syst. Saf. 51 (3) (1996) 229–240.
[2] M. Marseguerra, E. Zio, L. Podofillini, Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation, Reliab. Eng. Syst. Saf. 77 (2) (2002) 151–165.
[3] E. Zio, Review reliability engineering: Old problems and new challenges, Reliab. Eng. Syst. Saf. 94 (2) (2009) 125–141.
[4] C. Hu, P. Wang, B.D. Youn, W.R. Lee, Copula-Based Statistical Health Grade System against Mechanical Faults of Power Transformers, IEEE Transactions on Power Delivery 27 (4) (2012) 1809–1819.
[5] B.D. Youn, C. Hu, P. Wang, Resilience-Driven System Design of Complex Engineered Systems, Journal of Mechanical Design 133 (10) (2011). 101011(15).
[6] E. Myotyri, U. Pulkkinen, K. Simola, Application of stochastic filtering for lifetime prediction,, Reliab. Eng. Syst. Saf. 91 (2) (2006) 200–208.

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004

[7] F. Cadini, E. Zio, D. Avram, Model-based Monte Carlo state estimation for condition-based component replacement, Reliab. Eng. Syst. Saf. 94 (3) (2009) 752–758.
[8] J. Luo, K.R. Pattipati, L. Qiao, S. Chigusa, Model-based prognostic techniques applied to a suspension system, IEEE Trans. Syst. Man Cybern. A 38 (5) (2008) 1156–1168.
[9] N. Gebraeel, J. Pan, Prognostic degradation models for computing and updating residual life distributions in a time-varying environment,, IEEE Trans. Reliab. 57 (4) (2008) 539–550.
[10] N. Gebraeel, A. Elwany, J. Pan, Residual life predictions in the absence of prior degradation knowledge, IEEE Trans. Reliab. 58 (1) (2009) 106–117.
[11] X.S. Si, W. Wang, C.H. Hu, M.Y. Chen, D.H. Zhou, A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation,, Mech. Syst. Signal Process. 35 (1–2) (2013) 219–237.
[12] Schwabacher M., 2005, A survey of data-driven prognostics, in: Proceedings of AIAA Infotech@Aerospace Conference, Arlington, VA.
[13] X.S. Si, W. Wang, C.H. Hu, D.H. Zhou, Remaining useful life estimation-A review on the statistical data driven approaches, Eur. J. Oper. Res. 213 (1) (2011) 1–14.
[14] Wang T., Yu J., Siegel D., and Lee J., 2008, A similarity-based prognostics approach for remaining useful life estimation of engineered systems, in: Proceedings of the International Conference on Prognostics and Health Management, Denver, CO, Oct 6–9.
[15] P. Wang, B.D. Youn, C. Hu, A generic probabilistic framework for structural health prognostic and uncertainty management, Mech.l Syst. Signal Process. 28 (2012) 622–637.
[16] C. Hu, B.D. Youn, P. Wang, Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life, Reliab. Eng. Syst. Saf. 103 (2012) 120–135.
[17] Coble J.B., and Hines J.W., 2008, Prognostic algorithm categorization with PHM challenge application, in: Proceedings of the IEEE International Conference on Prognostics and Health Management, Denver, CO, Oct 6–9.
[18] Heimes F.O., 2008, Recurrent neural networks for remaining useful life estimation, in: Proceedings of the IEEE International Conference on Prognostics and Health Management, Denver, CO, Oct 6–9.
[19] V.T. Tran, H. Thom Pham, B. Yang, T. Tien Nguyen, Machine performance degradation assessment and remaining useful life prediction using proportional hazard model and support vector machine, Mech. Syst. Signal Process. 32 (2012) 320–330.
[20] D.A. Tobon-Mejia, K. Medjaher, N. Zerhouni, CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks,, Mech. Syst. Signal Process. 28 (2012) 167–182.
[21] Kozlowski J.D., Watson M.J., Byington C.S., Garga A.K., and Hay T.A., 2001, Electrochemical cell diagnostics using online impedance measurement, state estimation and data fusion techniques, in: Proceedings of the 36th Intersociety Energy Conversion Engineering Conference, Savannah, Georgia.
[22] Goebel K., Eklund N., and Bonanni P., 2006, Fusing competing prediction algorithms for prognostics, in: Proceedings of 2006 IEEE Aerospace Conference, New York.
[23] B. Saha, K. Goebel, S. Poll, J. Christophersen, Prognostics methods for battery health monitoring using a Bayesian framework, IEEE Trans. Instrum. Meas. 58 (2) (2009) 291–296.
[24] A. Heng, A.C.C. Tan, J. Mathewa, N. Montgomery, D. Banjevic, A.K.S. Jardine, Intelligent condition-based prediction of machinery reliability, Mech. Syst. Signal Process. 23 (5) (2009) 1600–1614.
[25] W. Caesarendra, A. Widodo, B.-S. Yang, Application of relevance vector machine and logistic regression for machine degradation assessment, Mech. Syst. Signal Process. 24 (4) (2010) 1161–1171.
[26] A. Widodo, B.-S. Yang, Application of relevance vector machine and survival probability to machine degradation assessment, Expert Syst. Appl. 38 (3) (2011) 2592–2599.
[27] Z. Tian, L. Wong, N. Safaei, A neural network approach for remaining useful life prediction utilizing both failure and suspension histories,, Mech. Syst. Signal Process. 24 (5) (2010) 1542–1555.
[28] Z.-H. Zhou, M. Li, Semisupervised regression with cotraining-style algorithms, IEEE Trans. Knowl. Data Eng. 19 (11) (2007) 1479–1493.
[29] Abdel Hady M.F., Schwenker F., and Palm G., Semi-supervised learning for regression with co-training by committee in: C. Alippi et al., (Eds.), Proceedings of the 19th International Conference on Artificial Neural Networks (ICANN 2009), LNCS 5768, Springer-Verlag, 2009, pp. 121–130.
[30] Z.-H. Zhou, M. Li, Semi-supervised learning by disagreement,, Knowl. Inf. Syst. 24 (3) (2010) 415–439.
[31] Byington C.S., Watson M., and Edwards D., 2004, Data-driven neural network methodology to remaining life predictions for aircraft actuator components, in: Proceedings of IEEE Aerospace Conference, March 6–13, vol. 6, pp. 3581–3589, http://dx.doi.org/10.1109/AERO.2004.1368175.
[32] Byington C.S., Watson M., and Edwards D., Dynamic signal analysis and neural network modeling for life prediction of flight control actuators,in: Proceedings of the American Helicopter Society 60th Annual Forum, Alexandria, VA: AHS, 2004.
[33] Liu J., Saxena A., Goebel K., Saha B., and Wang W., An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries, in: Proceedings of Annual Conference of the PHM Society, October 10–16, Portland, Oregon, 2010.
[34] S.J. Wu, N. Gebraeel, M.A. Lawley, Y. Yih, A neural network integrated decision support system for condition-based optimal predictive maintenance policy,, IEEE Trans. Syst. Man Cybern A – Syst. Hum. 37 (2) (2007) 226–236.
[35] M. Hagan, H. Demuth, M. Beale., Neural Network Design, PWS Publishing, Boston, MA, 1996.
[36] J. Park, I.W. Sandberg, Approximation and radial-basis function networks, Neural Comput. 5 (1993) 305–316.
[37] F. Schwenker, H. Kestler, G. Palm, Three learning phases for radial basis function networks,, Neural Netw. 14 (4–5) (2001) 439–458.
[38] P.D. McFadden, J.D. Smith, Model for the vibration produced by a single point defect in a rolling element bearing, J. Sound Vib. 96 (1) (1984) 69–82.
[39] Y.F. Wang, P.J. Kootsookos, Modeling of low shaft speed bearing faults for condition monitoring, Mech. Syst. Signal Process. 12 (3) (1998) 415–426.
[40] Saxena A., and Goebel K., Damage propagation modeling for aircraft engine run-to-failure simulation,in: Proceedings of the IEEE International Conference on Prognostics and Health Management, Denver, CO, Oct 6–9, 2008.
[41] J.F. Bercher, C. Vignat, Estimating the entropy of a signal with applications, IEEE Trans. Signal Process. 48 (6) (2000) 1687–1694.
[42] Y. Shao, K. Nezu, Prognosis of remaining bearing life using neural networks, Proc. Inst. Mech. Eng. I, J. Syst. Control Eng. 214 (3) (2000) 217–230.
[43] Tian X., Cooling fan reliability, failure criteria, accelerated life testing, modeling, and quantification, in: Proceedings of the IEEE Annual Reliability and Maintainability Symposium, Newport Beach, CA, Jan 23–26, 2006.
[44] R. Burger, Cooling Tower Technology—Maintenance, Updating And Rebuilding, Fairmont Press, Lilburn, GA, 1995.
[45] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Mach. Learn. 39 (2–3) (2000) 103–134.

Please cite this article as: C. Hu, et al., A co-training-based approach for prediction of remaining useful life utilizing both failure and suspension data, Mech. Syst. Signal Process. (2015), http://dx.doi.org/10.1016/j.ymssp.2015.03.004