

## Research Article

# Fault Prognostic Based on Hybrid Method of State Judgment and Regression

Xiaobin Li,<sup>1,2</sup> Jiansheng Qian,<sup>1</sup> and Gai-ge Wang<sup>2</sup>

<sup>1</sup> School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221008, China

<sup>2</sup> School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116, China

Correspondence should be addressed to Jiansheng Qian; [qianjsh@cumt.edu.cn](mailto:qianjsh@cumt.edu.cn)

Received 19 July 2013; Accepted 29 September 2013

Academic Editor: Amir H. Gandomi

Copyright © 2013 Xiaobin Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fault prognostic is one of the most important problems in equipment health management system. This paper presents a hybrid method of mixture of Gaussian hidden Markov model (MG-HMM) and fixed size least squares support vector regression (FS-LSSVR) for fault prognostic. The system is established based on three parts. The first part trains the MG-HMM and FS-LSSVR model. According to the known samples, several MG-HMM models can be learned based on expectation maximization (EM) algorithm. Then, the forward variables can be calculated based on these MG-HMM models. Based on these forward variables, the corresponding FS-LSSVR models are built. All the MG-HMM models and corresponding FS-LSSVR models are combined into a model library. The second part recognizes the unknown sample based on the model library. This part obtains the MG-HMM model and FS-LSSVR model by maximization likelihood calculation between the unknown sample and MG-HMM models. The third part of the system calculates the forward variables based on the MG-HMM obtained from the second part. These forward variables are inputted into the corresponding FS-LSSVR model to compute the remaining useful life (RUL) of the unknown sample. Finally, we carry out experiments on benchmark data set to verify the proposed method. The results illustrate the effectiveness of the hybrid method.

## 1. Introduction

RUL is one of the most important problems in many application areas such as condition based maintenance (CBM) [1], fault prognostics (FP) [2], and prognostics and health management (PHM) system [3]. Obviously, the RUL of a system or a component is a random variable. Three techniques are applied to estimate the RUL: life cycle model, expert knowledge system, and data-driven model [4]. The data-driven model compromises the merits of adaptability, low cost, and little expert knowledge. For these reasons, it has been widely concerned in recent years.

Many artificial intelligence (AI) techniques have been applied to fault prognostic. Expert system (ES) is a computer system which consists of a knowledge database and an inference mechanism. The knowledge database contains domain knowledge for the solution of problems. This approach is not best for situations where there is a lack of expert knowledge

[5]. ES cannot handle new situations not covered explicitly in its knowledge database. Fuzzy logic can represent uncertainty of complexity system while it is not feasible where there is no suitable membership function [6]. Artificial Neural Network (ANN) can simulate the structures and functions of biological neural networks. This approach can capture potential knowledge from input patterns. However there is no standard method to determine the structure of the network [7]. Support vector machine (SVM) projects feature space into a higher dimensional space by a kernel function and finds an optimized separation surface in the higher dimensional space. This approach can achieve better generalization capability than ANN while there is no standard kernel function selection method for SVM [8].

Hidden Markov model (HMM) has demonstrated its superior performance in lots of application areas, such as speech recognition and gesture recognition. It is more suitable for modeling on stationary stochastic signal. Meanwhile,

HMM is also inspected and applied into fault diagnosis or fault prognostics [9]. Comparing with many other RUL estimation methods, a distinct advantage of the HMM model is that HMM model can give an intelligible explanation. Bunks et al. firstly put forward a HMM model built on the Westland helicopter gear monitoring data, and then the RUL of gear is estimated based on the HMM model [10]. In this paper, the authors estimate each of the 68 operating conditions with a different eight-dimensional Gaussian model. At the same time, they make hypothesis that the health state could not revert once it departs from the current health state during its operation. Baruah and Chinnam apply HMM to model the metal cutting tools [11, 12]. The proposed method can simultaneously meet the requirements of fault diagnosis and fault prognostics. To describe a three-state transition procedure, a two-mixture Gaussian model is used for modeling. Based on the previous model, once the previous state transition time is known, the next state transition time can be estimated. Camci and Chinnam make a thorough inspection on drill bit health evaluation [13]. They define the drill bit health level as drilling holes number. Each drilling history time series can be used to train a separate HMM model. These HMM models are used to estimate the drill bit health in time. Then a hierarchy HMM is established to model the state transition relation in which the upper HMM model describes the drill bit health state transition relationship while the lower HMM model describes the drilling time series emitted from one state in the upper HMM model. Once the drill bit health is determined, the RUL is calculated based on the probability transition matrix in the upper HMM model. Ocaik et al. make a decomposition on the bearing vibration signals by using wavelet packet technique. The  $N$ th layer node data is used as feature vector for HMM training. If the bearing state need to be judged, the HMM models will be applied into likelihood calculation. The experimental results found that, as the bearing wears, the likelihood gradually drops [14]. Tai et al. apply HMM model into condition monitoring of nozzle [15]. Zhou et al. combine the HMM model and belief rule base under a variable circumstance, where belief rules are used to model the dynamic environment. Eventually, fault diagnosis and fault prognostics are implemented for the complex system based on this mixture model [16]. Tobon-Mejia et al. establish the models on the bearing data with mixture of gaussian hidden Markov model (MG-HMM). Once a sequence classification is determined, a graph-based path traversal algorithm is carried on to estimate the RUL of bearing [17]. Liu et al. present a hybrid method of HMM and LSSVR to predict the RUL. The LSSVR is used to predict the future features based on the past samples and the HMM is used to judge the future state according to the future predicted features [18].

Motivated by the above approaches for fault prognostic, in this paper, we present a hybrid method of state judgment and regression. The state judgment is implemented by MG-HMM model. The MG-HMM not only can make condition recognition but also provides detailed health indexes for unknown sample. The regression is implemented by FS-LSSVR. FS-LSSVR is a version of SVM which is more suitable for large scale application. The RUL can be calculated by

the FS-LSSVR model built on the health indexes obtained from MG-HMM. We also compare the RUL prediction performance between artificial neural network (ANN) and FS-LSSVR. The experimental results show that the hybrid method is effective for fault prognostic.

This paper is organized as follows. Section 2 introduces the MG-HMM and its training. Section 3 introduces the FS-LSSVR model which is fit for large scale problem. Section 4 gives a detailed description for left-right HMM (LR-HMM) and system architecture which is used for fault diagnosis and fault prognostic. The experiments are carried out on a benchmark dataset to illustrate the effectiveness of the HMM training method in Section 5. The last section is the conclusion of this paper.

## 2. HMM Model and Its Training

**2.1. HMM Model.** Discrete hidden Markov model (DHMM) is a doubly stochastic process [19], and a DHMM model usually contains the following elements:

- (1)  $S = \{s_1, s_2, \dots, s_N\}$ , which is a finite set of states where each element means a distinct state,
- (2)  $V = \{v_1, v_2, \dots, v_M\}$ , which is a set of output symbols,
- (3)  $A$ , which is state transition probability matrix where  $a_{ij} = P[q_{t+1} = s_j | q_t = s_i]$ ,  $1 \leq i, j \leq N$ ,
- (4)  $B = \{b_j(k)\}$ , which is an observation value probability distribution, where  $b_j(k) = P[v_k | q_t = s_j]$ ,  $1 \leq j \leq N$ ,  $1 \leq k \leq M$ ,
- (5)  $\pi = \{\pi_i\}$ , which is an initial state probability distribution, where each element means a probability of the initial state.  $\pi_i = P(q_1 = s_i)$ ,  $1 \leq i \leq N$ .

Usually, a more compact model  $\lambda = (A, B, \pi)$  is used to represent a HMM model for there is an implication definition  $N$  and  $M$  in  $A$  and  $B$ .

DHMM only considers the case when the observations were discrete symbols chosen from a finite alphabet. There are some disadvantages for this method if the observations are continuous signals. Although it is possible to quantize such continuous signals via codebooks and so forth, there might be serious degradation associated with such quantization. Hence, it would be advantageous to use MG-HMM with continuous observation densities.

The continuous observation under a special state can be described as a probability density function. Usually, mixture Gaussian distribution probability density function is used to describe the probability density function:

$$b_j(O) = \sum_{m=1}^M c_{jm} N(O, \mu_{jm}, U_{jm}), \quad 1 \leq j \leq N, \quad (1)$$

where  $O$  is the vector being modeled,  $c_{jm}$  is the mixture coefficient for the  $m$ th mixture in state  $j$ , and  $N$  is a Gaussian distribution function, with mean vector  $\mu_{jm}$  and covariance

matrix  $U_{jm}$  for the  $m$ th mixture component in state  $j$ . The mixture coefficient  $c_{jm}$  satisfies the stochastic constraint

$$\sum_{m=1}^M c_{jm} = 1, \quad 1 \leq j \leq N, \quad (2)$$

$$c_{jm} \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq m \leq M.$$

So that the probability density function is properly normalized; that is,

$$\int_{-\infty}^{\infty} b_j(x) dx = 1, \quad 1 \leq j \leq N. \quad (3)$$

**2.2. Model Training.** DHMM parameter estimation problem can be defined as how to adjust unknown model parameters  $\lambda = (A, B, \pi)$  to maximize  $P(O | \lambda)$ .

Baum-Welch (BW) or gradient descent algorithm can be applied to solve this problem. Now, a brief description of BW algorithm is provided here.

Firstly, a variable  $\xi_t(i, j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda)$  is defined to represent a transition probability from state  $s_i$  at time  $t$  to state  $s_j$  at time  $t + 1$ .

By definition, it can be deduced that

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \quad (4)$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}.$$

Hence, a probability under state  $s_i$  at time  $t$  can be defined as

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j). \quad (5)$$

Therefore, the three parameters of HMM model can be estimated. The initial probability distribution can be calculated as

$$\bar{\pi}_i = \gamma_1(i). \quad (6)$$

The transition probability distribution can be calculated as

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \quad (7)$$

The emission probability distribution can be calculated as

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}. \quad (8)$$

Given an observation sequence  $O = O_1, O_2, \dots, O_T$ , Baum-Welch method may be used to estimate  $\lambda$  such that  $P(O | \lambda)$  is locally maximized. Firstly, three variables are defined as follows.

Forward variable  $\alpha_t(i) = P(O_1, O_2, \dots, O_t, q_t = s_i | \lambda)$ , that is, the probability of the partial observation sequence,  $O_1, O_2, \dots, O_t$ , and state  $s_i$  at time  $t$ , given the model  $\lambda$ .

Backward variable  $\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T | q_t = s_i, \lambda)$ , that is, the probability of the partial observation sequence from  $t + 1$  to the end, given state  $s_i$  at time  $t$  and model  $\lambda$ .

$\omega_t(i, j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda)$ , the probability of being in state  $s_i$  at time  $t$ , and  $s_j$  at time  $t + 1$ , given the model  $\lambda$  and the observation sequence  $O$ .

The detailed model training steps of MG-HMM are listed as follows.

**Step 1.** Some variables are firstly defined: state number  $N$ , Gaussian mixture component number  $M$ , initial state distribution with random initialization, state transition probability distribution  $A$  with random initialization and the other parameters needed to construct observation symbol probability distribution in state  $j$  with random initialization. Of course the random initialization should be satisfied with the stochastic constraint.

**Step 2.** Calculate  $\alpha_t(i)$ . This variable can be solved by dynamic programming algorithm. The solving process is as follows:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq j \leq N,$$

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad (9)$$

$$1 \leq j \leq N, \quad 1 \leq t \leq T - 1.$$

**Step 3.** Calculate  $\beta_t(i)$ . The solving process is as follows:

$$\beta_T(i) = 1, \quad 1 \leq j \leq N,$$

$$\beta_t(j) = \sum_{i=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(i), \quad 1 \leq j \leq N, \quad (10)$$

$$t = T - 1, T - 2, \dots, 1.$$

**Step 4.** Calculate  $\omega_t(i, j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda)$  for each time  $t$ :

$$\omega_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \quad (11)$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}$$

**Step 5.** Calculate  $\gamma_t(i) = \sum_{j=1}^N \omega_t(i, j)$  for each time  $t$ .

**Step 6.** Calculate updated initial probability distribution  $\pi$  and state transition probability distribution  $A$ :

$$\bar{\pi}_i = \gamma_1(i),$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \omega_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \quad (12)$$

*Step 7.* For each time  $t$ , state  $j$  and the  $k$ th component probability in mixture Gaussian distribution can be expressed as

$$\gamma_t(j, k) = \left[ \frac{\alpha_t(j) \beta_t(j)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)} \right] \left[ \frac{c_{jk} N(O_t, \mu_{jk}, U_{jk})}{\sum_{m=1}^M c_{jm} N(O_t, \mu_{jm}, U_{jm})} \right]. \quad (13)$$

*Step 8.* Estimate the parameters in observation probability density function:

$$\begin{aligned} \bar{c}_{jk} &= \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j, k)}, \\ \bar{\mu}_{jk} &= \frac{\sum_{t=1}^T \gamma_t(j, k) O_t}{\sum_{t=1}^T \gamma_t(j, k)}, \\ \bar{U}_{jk} &= \frac{\sum_{t=1}^T \gamma_t(j, k) (O_t - \mu_{jk}) (O_t - \mu_{jk})'}{\sum_{t=1}^T \gamma_t(j, k)}. \end{aligned} \quad (14)$$

*Step 9.* The procedure may end or jump to Step 2 according to the given threshold.

The above model training algorithm can only be applied for a single observation sequence. In practical application, in order to obtain a more reliable estimation of parameters, one has to use multiple observation sequences.

For given multiple sequences  $O = [O^{(1)}, O^{(2)}, \dots, O^{(z)}]$  and each sequence  $O^{(z)} = [O_1^{(z)}, O_2^{(z)}, \dots, O_{T_z}^{(z)}]$ , the goal of model training is  $P(\lambda | O) = \max_{\lambda} \prod_{z=1}^z P(O^{(z)} | \lambda)$ . We can adaptively modify the above algorithm to confirm with the multiple sequences. First of all, the likelihood  $P(O^{(z)} | \lambda) = \sum_{i=1}^N \alpha_T^{(z)}(i)$  of each single sequence is calculated. Hence, the reestimation of parameters should consider the contribution weight of each sequence.

### 3. FS-LSSVR

SVM has been widely used in many areas [20]. Least squares support vector machine (LSSVM) [21] is a least squares version of SVM for classification problems. The solution of LSSVM follows directly from solving a set of linear equations. Furthermore, the support values in LSSVM are proportional to the errors.

**3.1. LSSVM.** Given a sample set  $\{y_k, x_k\}_{k=1}^N$  where  $x_k \in R^n$  is the  $k$ th input features and  $y_k \in R$  is the  $k$ th label, LSSVM can be described as follows [21]:

$$\begin{aligned} \min_{w, b, e} \quad & J = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{k=1}^N e_k^2 \\ \text{subject to} \quad & y_k [w^T \phi(x_k) + b] = 1 - e_k, \\ & k = 1, \dots, N. \end{aligned} \quad (15)$$

The solution to the constrained optimization problem follows from the Lagrangian:

$$L(w, b, e; \alpha) = J(w, b, e) - \sum_{k=1}^N \alpha_k \{y_k [w^T \phi(x_k) + b] - 1 + e_k\}. \quad (16)$$

Here, the variable  $\alpha_k$  is Lagrange multiplier. The problems can be solved by the following equation:

$$\begin{aligned} \frac{\partial L}{\partial w} = 0 &\longrightarrow w = \sum_{k=1}^N \alpha_k y_k \phi(x_k), \\ \frac{\partial L}{\partial b} = 0 &\longrightarrow \sum_{k=1}^N \alpha_k y_k = 0, \\ \frac{\partial L}{\partial e_k} = 0 &\longrightarrow \alpha_k = \gamma e_k, \quad k = 1, \dots, N, \\ \frac{\partial L}{\partial \alpha_k} = 0 &\longrightarrow y_k [w^T \phi(x_k) + b] - 1 + e_k = 0, \\ &k = 1, \dots, N. \end{aligned} \quad (17)$$

Similar to the standard SVM, the  $w$  and  $\phi(x_i)$  do not need to be solved. A linear KKT system can be established in place of the two order optimization system to eliminate  $w$  and  $e$ :

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \Omega + \gamma^{-1} I_N \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix}. \quad (18)$$

In the above linear equation, there are  $Y = [y_1; \dots; y_N]$  and  $1_v = [1; \dots; 1]$ ,  $e = [e_1; \dots; e_N]$ , and  $\alpha = [\alpha_1; \dots; \alpha_N]$ . Meanwhile, according to the Mercer permission condition, the kernel matrix  $\Omega \in R^{N \times N}$  can be written as

$$\Omega_{ij} = \phi(x_i)^T \phi(x_j) = K(x_i, x_j). \quad (19)$$

The solution of classification problem is as follows:

$$y(x) = \text{sign} \left[ \sum_{i=1}^N \alpha_i y_i K(x, x_i) + b \right]. \quad (20)$$

**3.2. FS-LSSVR.** Solving the LS-SVM requires the resolution for all samples, which is practical when the input space dimension is larger than sample size. However, when the sample size is very large, it is impossible to solve these questions by traditional LS-SVM method [22]. For example, the benchmark data set used in Section 5 to finish fault prognostic contains about 200 samples, and the total size is above 20000. For this case, LS-SVR needs to be adjusted to fit for large scale problems. FS-LSSVR is presented to solve this problem [22]. It makes use of the Nyström approximation but estimates the model in the primal within the LS-SVM setting [23]. Instead of a random subset, a subset selection method based upon quadratic Renyi entropy was proposed.

(1) *Nyström Approximation in Dual Space.* Nonlinear map  $\phi$  can be explicitly represented by eigenvalue decomposition of

kernel matrix  $\Omega$  and kernel function  $K(x, x_j)$ . For probability density  $p(x)$ , there is

$$\int K(x, x_j) \varphi_i(x) p(x) dx = \lambda_i \varphi_i(x_j). \quad (21)$$

$\phi$  is represented as follows:

$$\phi = [\sqrt{\lambda_1} \varphi_1, \sqrt{\lambda_2} \varphi_2, \dots, \sqrt{\lambda_{nh}} \varphi_{nh}]. \quad (22)$$

Here, the  $\varphi$  is eigenvalue function.

The above problem can be transformed to an eigenvalue problem on the sample data set:

$$\frac{1}{N} \sum_{k=1}^N K(x_k, x_j) u_i(x_k) = \lambda_i^{(s)} u_i(x_j). \quad (23)$$

Then, the eigenvalue  $\lambda_i$  and eigenvalue vector  $u_i$  can be approximated by the eigenvalue and eigenvalue vector:

$$\hat{\lambda}_i = \frac{1}{N} \lambda_i^{(s)}, \quad \hat{\varphi}_i = \sqrt{N} u_i. \quad (24)$$

The nonlinear map is estimated as follows:

$$\hat{\phi}_i(x) = \frac{N}{\sqrt{\lambda_i^{(s)}}} \sum_{k=1}^N u_{ki} K(x_k, x^{(v)}). \quad (25)$$

(2) *Sparse Approximation for Subset*. Small proportion of all samples selected to approximate all samples is the core idea of FS-LSSVR algorithm. The standard of sample selection is of great importance. Entropy maximization is a well-defined criterion for subset selection. Renyi entropy can be considered as one of the criteria:

$$H_R = -\log \int p(x)^2 dx. \quad (26)$$

It can be approximated as

$$\int \hat{p}(x)^2 dx = \frac{1}{N^2} \mathbf{1}^T \mathbf{\Omega} \mathbf{1}. \quad (27)$$

(3) Based on the analysis in (1) and (2), the algorithm can be listed as follows.

*Step 1.* Randomly select a subset with  $M$  samples from the original data set  $\{y_k, x_k\}_{k=1}^N$ . The two-order Renyi entropy is calculated for the selected samples. Then, the subset with maximization Renyi entropy is determined.

*Step 2.* A small kernel matrix  $\Omega_M$  is constructed based on the selected samples.

*Step 3.* Compute the eigenvalue  $\lambda_i^{(s)}$  and eigenvalue vector  $u_i$  on  $\Omega_M$ .

*Step 4.* Nonlinear map  $\hat{\phi}(x_i)$  is calculated for all samples ( $i = 1, \dots, N$ ).

*Step 5.* The final regression problem is solved based on the above steps.

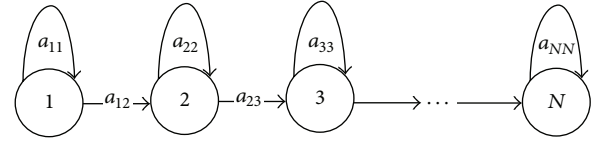


FIGURE 1: LR-HMM Model for RUL prediction.

## 4. System Framework for RUL Prediction

*4.1. LR-HMM.* Here LR-HMM model [7] is proposed for RUL prediction as shown in Figure 1.

The health status of system discussed here is  $S = \{1, 2, 3, \dots, N\}$ . The health status is a oneway and irreversible. From Section 2, we know that the forward variable  $\alpha_t(i) = P(O_1, O_2, \dots, O_t, q_t = S_i | \lambda)$  represents the probability of the partial observation sequence,  $O_1, O_2, \dots, O_t$ , and state  $S_i$  at time  $t$ , given the model  $\lambda$ . This definition is used to describe the health status of system. The calculation of  $\alpha_t(i)$  is given in (9).

*4.2. System Framework.* The system is established based on three parts as shown in Figure 2. The first part trains the MG-HMM and FS-LSSVR model. According to the known samples, several MG-HMM models can be trained based on expectation maximization algorithm. Then, the forward variables of each sample can be calculated based on these MG-HMM models. The corresponding FS-LSSVR model is established with these forward variables. The MG-HMM models and corresponding FS-LSSVR models are constructed into a model library. The second part of the system is to recognize the unknown sample based on the model library. This part determines the MG-HMM model and FS-LSSVR with the maximization likelihood.

The third part of the system calculates the forward variables based on the MG-HMM obtained from the second part. These forward variables are put into the corresponding FS-LSSVR model to compute the RUL of the unknown sample.

## 5. Experiments

*5.1. RUL Evaluations Metrics.* Fault prognostic has its inherent particularity contrary to fault diagnosis. Therefore, Saxena et al. present some new evaluation metrics on fault prognostics [24–27]. After making a thorough inspection of these metrics, this paper further proposes two metrics for fault prognostics metrics: MA $\alpha$  and  $\alpha$ -Nmap. We consider the series of evaluation metrics that could effectively measure algorithm performance in real-time RUL assessment.

First of all, a variable  $r^f(t)$  is defined as the actual RUL when the system is in  $t$  moment, while variable  $r^p(t)$  is defined as the prediction RUL at the same time. Hence, the prediction error percentage is represented as

$$PE(t) = \frac{|r^p(t) - r^f(t)|}{r^f(t)}. \quad (28)$$



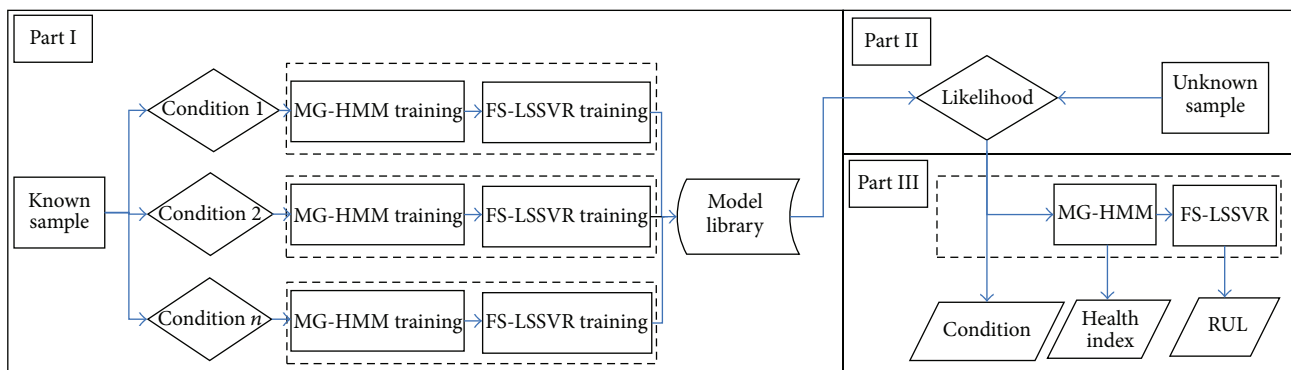


FIGURE 2: System framework.

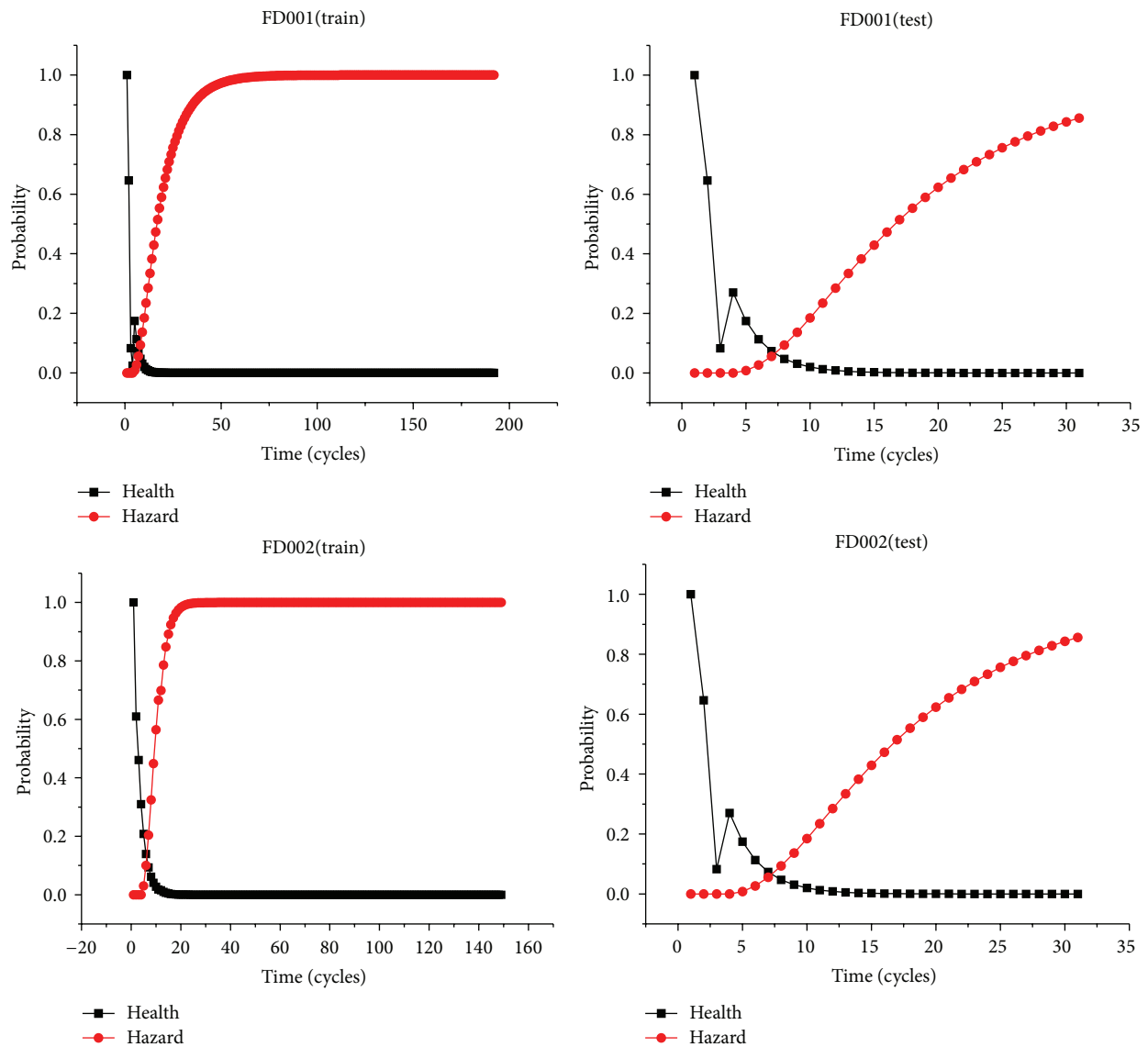


FIGURE 3: Continued.

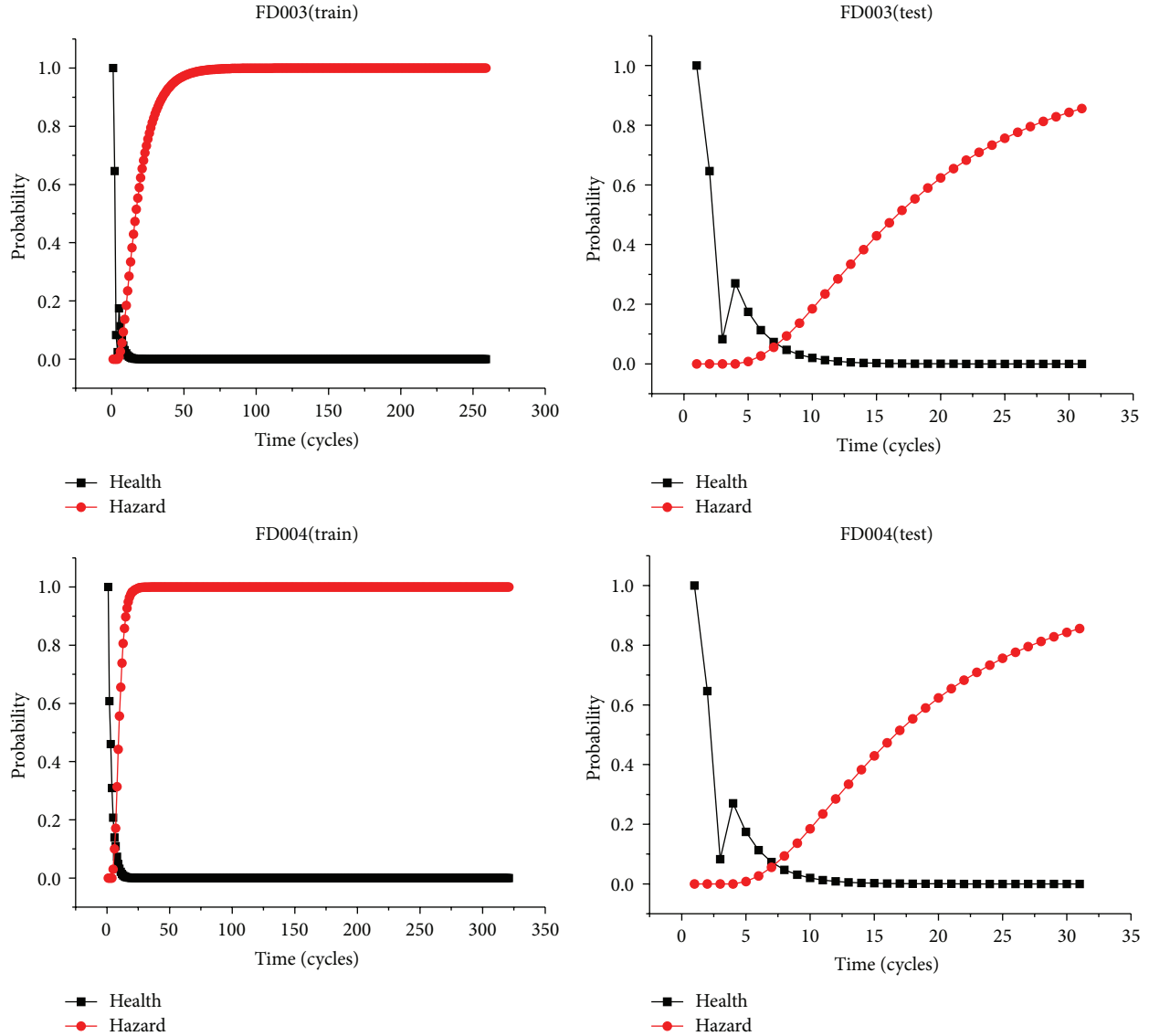


FIGURE 3: Health indexes for several samples.

Obviously there is  $PE \geq 0$ . When the PE is equal to zero, there will be a perfect prediction.

(I) *MAPE*. Assumes the prediction time start that from  $t = 1$  to the final failure moment  $t = N$ . Based on the prediction error percentage metrics, there are several evaluation metrics listed below

$$MAPE = \frac{1}{N} \sum_{t=1}^N PE(t). \quad (29)$$

The metrics measure the average prediction precision along with the time axis.

(II) *BIAS*. Consider the following:

$$BIAS = \sqrt{\frac{1}{N-1} \sum_{t=1}^N (PE(t) - MAPE)^2}. \quad (30)$$

The metrics measure the prediction bias along with the time axis.

(III)  $\alpha(t)$ ,  $\alpha \in [0, 1]$ . If there is  $[1 - \alpha]r^f(t) \leq r^p(t) \leq [1 + \alpha]r^f(t)$ , there will be  $\alpha(t) = \text{true}$ , otherwise  $\alpha(t) = \text{false}$ . The metrics gives a Boolean determination whether the prediction time is in a certain earlier or later range away from actual time.

(IV) *MA $\alpha$* . Consider the following:

$$MA\alpha = \frac{1}{N} \left( \sum_{t=1}^N I(\alpha(t)) \right), \quad \alpha \in [0, 1]; \quad (31)$$

where there is  $\alpha(t) = \text{true}$  there will be  $I(\alpha(t) = \text{true}) = 1$ , otherwise,  $I(\alpha(t) = \text{true}) = 0$ . In fact, these metrics makes a simple statistics of  $\alpha(t)$ . The great advantage of the metrics

TABLE 1: C-MAPSS data set.

Data Set	Train trajectories	Test trajectories	Conditions	Fault modes
FD001	100	100	One (sea level)	One (HPC Degradation)
FD002	260	259	Six	One (HPC degradation)
FD003	100	100	One	Two (HPC degradation, fan degradation)
FD004	248	249	Six	Two (HPC degradation, fan degradation)

is that they do not need to consider specific prediction time point.

(V)  $\alpha$ -Nmap. Furthermore, we specify  $\lambda = 1/k$  and  $\alpha = \lambda m$  where there is  $m = \{1, 2, \dots, k\}$ . Hence, the  $\alpha$ -Nmap is represented as

$$\alpha\text{-Nmap} = \frac{1}{k} \sum_{m=1}^k M A \alpha(m\lambda). \quad (32)$$

The metrics mean the whole prediction correct percentage when the allowable prediction range changes in the interval  $[0, 1]$ . Therefore, there is  $\alpha$ -Nmap  $\in [0, 1]$ . Furthermore, the best solution is obtained when  $\alpha$ -Nmap = 1.

(VI) C. The metrics indicate the convergence performance of prediction. The detailed description can be provided as follows.

$$\begin{aligned} x_c &= \frac{(1/2) \sum_{t=1}^{N-1} ((t+1)^2 - t^2) PE(t)}{\sum_{t=1}^N PE(t)}, \\ y_c &= \frac{(1/2) \sum_{t=1}^{N-1} PE(t)^2}{\sum_{t=1}^N PE(t)}, \\ C &= \sqrt{x_c^2 + y_c^2}. \end{aligned} \quad (33)$$

**5.2. Data Set.** (C-MAPSS) Commercial modular aeropropulsion system simulation, [24] is a tool for simulating a realistic large commercial turbofan engine. The software simulates an engine model of the 90000 lb thrust class and an atmospheric model capable of simulating operations at (i) altitudes ranging from sea level to 40000 ft, (ii) March numbers from 0 to 0.90, and (iii) sea-level temperatures from  $-60$  to  $103^\circ\text{F}$ . Several dataset generated by C-MAPSS has been uploaded on NASA website. The description of the dataset is shown in Table 1.

There are 26 columns in these data sets. Column one represents the unit number. Column two is the time of unit running. Columns 3–5 are the different unit operation conditions. The remaining columns are the 21 sensor measurements. The software calculates the health index according to four parameters of the 21 variables. But the researchers could not find these parameters explicitly.

The train trajectories contain all sensor measurements of one unit during its whole life. Otherwise, the test trajectories only provide part of all measurements of one unit. That is, the dataset are truncated at an appropriate time before its failure. The data set is divided into four training sets and

four test sets. In the experiments, we used FD001 and FD002 train trajectories as training sets and FD001 and FD002 test trajectories as test set.

In order to train the MG-HMM model on data, we firstly transform the 24 sensor measurements (including 3 columns working conditions and 21 columns other sensor measurements) into values in  $(0, 1)$  range by Sigmoid function  $x = 1/(1 + \exp(x))$ . For the considerations of performance and precision, here we choose the HMM model with five states and two Gaussian mixture models. The model training, unknown sample recognition, and RUL prediction process are listed as follows.

- (1) Four individual MG-HMM models are trained on the four data sets with different work conditions.
- (2) Health index of different data sets is calculated based on the corresponding MG-HMM.
- (3) FS-LSSVR models corresponding to the MG-HMM are built based on health index, running cycle  $t$ , and RUL of training data set.
- (4) Online recognition for unknown sample is done based on MG-HMM. In other words, the algorithm determines the unknown sample as the emission of the MG-HMM which has the maximization likelihood on the unknown sample.
- (5) Health index of the time  $t$  is calculated based on the MG-HMM obtained in (4).
- (6) Based on the time cycle  $t$  and health index, the FS-LSSVR corresponding to the MG-HMM obtained in (4) is used to predict the RUL.

**5.3. Condition Recognition.** These MG-HMM models are used to judge the work condition of unknown sample. The judgment results are listed in the first row of Table 2. From Table 2, we find the recognition accuracy on FD001 and FD002 is very high, while the recognition accuracy of FD003 and FD004 is very low. The reason is that the FD001 and FD002 only contain one type fault, that is, HPC degradation. However, the FD003 and FD004 contain two type faults; one is HPC degradation and the other is fan degradation. Furthermore, we give a deeper inspector for the dataset. Then we find that FD003 operates under the same condition as FD001 and FD004 operates under the same condition as FD002. Then we adjust the algorithm with respect to the work condition. The algorithm is considered to do right recognition if it judges the FD003 as the emission of MG-HMM1 which is trained from FD001. Similarly, the algorithm is thought to do right recognition if it judges the FD004 as



TABLE 2: Accuracy of condition recognition (%).

	FD001	FD002	FD003	FD004
Before adjustment	99.06	77.69	0.72	22.29
After adjustment	—	—	100.00	99.87

the emission of MG-HMM2 which is trained from FD002. After this adjustment, the accuracy of condition recognition for FD003 is improved from 0.72% to 100%. Meanwhile, the accuracy of condition recognition for FD004 is improved from 22.29% to 99.87%.

**5.4. Health Indexes.** We apply the various MG-HMM models obtained from the training data set into health index calculation on the unit 1 of training data set and testing data set. The health indexes are illustrated in Figure 3. In Figure 3, the health state is the first state of MG-HMM and the hazard state is the final state. We find the probability of health decreases as time goes on. Meanwhile, the probability of hazard increases as time goes on. The situation is in line with the actual working condition of the traditional equipment.

**5.5. RUL Assessment.** Once health indexes of unknown samples are calculated based on MG-HMM models, the RUL can be predicated based on FS-LSSVR models. Here we also make a comparison between ANN and FS-LSSVR. The reason we choose ANN is that ANN is a good nonlinear regression tool in machine learning area. The comparison results are listed in Table 3. From MAPE metric, we find that the FS-LSSVR can make a higher accuracy RUL prediction than ANN. Based on BIAS metric, we find the deviation of RUL prediction based on FS-LSSVR is lower than ANN. From the comparison result of MA $\alpha$  and  $\alpha$ -Nmap, we find the FS-LSSVR can make more near RUL prediction than ANN. Also the FS-LSSVR converges to the failure point faster than ANN.

## 6. Conclusions

In this paper we present a hybrid method of MG-HMM and the FS-LSSVR for fault prognostic. The system contains three parts. The first part is responsible for training the MG-HMM models and FS-LSSVR models. The second part of the system recognizes the unknown sample based on the model library established in the first part. The third part of the system calculates the forward variables based on the MG-HMM obtained from the second part. These forward variables are inputted into the corresponding FS-LSSVR model to compute the RUL of the unknown sample.

Furthermore, after making a thorough inspection for former metrics of RUL assessment, this paper proposes two novel metrics for fault prognostics metrics: MA $\alpha$  and  $\alpha$ -Nmap. From the comparison result of MA $\alpha$  and  $\alpha$ -Nmap, we find the FS-LSSVR can make more near RUL prediction than ANN.

One interesting direction for future work is to investigate other state judgment techniques. The techniques can be developed based on pattern recognition techniques. Among

TABLE 3: RUL assessment comparison between ANN and FS-LSSVR.

Regression	Dataset	MAPE	BIAS	MA $\alpha$	$\alpha$ -Nmap	C
ANN	FD001	0.43	0.43	0.29	0.07	6745.52
	FD002	0.48	0.70	0.29	0.07	8415.49
	FD003	0.44	0.36	0.24	0.07	10753.88
	FD004	0.49	0.53	0.23	0.07	11270.31
FS-LSSVR	FD001	0.38	0.33	0.37	0.07	5779.59
	FD002	0.44	0.55	0.33	0.07	7414.83
	FD003	0.40	0.27	0.30	0.07	10322.90
	FD004	0.47	0.40	0.27	0.07	11066.95

these techniques the methods which can generate probability of health type are more preferred.

Another interesting direction for future work is to explore different regression methods. For example, the deep neural network (DNN) [25] and extreme learning machine (ELM) [28] are two newly developing machine learning algorithms. Maybe there is a better RUL prediction accuracy if the system framework can replace the FS-LSSVR with DNN or ELM. However, the calculation performance for large scale application of the two algorithms is still open issue, as we have considered in this paper.

## Acknowledgments

The authors wish to thank the anonymous reviewers and the editors for their constructive suggestions. This paper is supported by the following funding: National High Technology Research and Development Program of China (863 Program) (2012AA062103); Jiangsu Province Innovation Fund prospective Joint Research Project (BY2012081); Jiangsu Province Science and Technology Achievement Transformation Project (BA2010058); Jiangsu Normal University key foundation (10XLA13).

## References

- [1] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.
- [2] A. Heng, S. Zhang, A. C. C. Tan, and J. Mathew, "Rotating machinery prognostics: state of the art, challenges and opportunities," *Mechanical Systems and Signal Processing*, vol. 23, no. 3, pp. 724–739, 2009.
- [3] M. Pecht, J. Mathew, L. Ma, A. Tan, M. Weijnen, and J. Lee, *A Prognostics and Health Management for Information and Electronics-Rich Systems Engineering Asset Management and Infrastructure Sustainability*, Springer, London, UK, 2012.
- [4] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation—a review on the statistical data driven approaches," *European Journal of Operational Research*, vol. 213, no. 1, pp. 1–14, 2011.
- [5] J. L. Godwin, P. Matthews, and C. Watson, "Classification and detection of electrical control system faults through SCADA data analysis," *Chemical Engineering Transactions*, vol. 33, pp. 985–990, 2013.

- [6] B. W. Lo, R. L. Macdonald, A. Baker, and M. A. Levine, "Clinical outcome prediction in aneurysmal subarachnoid hemorrhage using Bayesian neural networks with fuzzy logic inferences," *Computational and Mathematical Methods in Medicine*, vol. 2013, Article ID 904860, 10 pages, 2013.
- [7] Z. Zhang, Y. Wang, and K. Wang, "Intelligent fault diagnosis and prognosis approach for rotating machinery integrating wavelet transform, principal component analysis, and artificial neural networks," *The International Journal of Advanced Manufacturing Technology*, vol. 68, no. 1–4, pp. 763–773, 2013.
- [8] X. Zhou, Z. Xiang, M. Liu, and J. Xiang, "A multiwavelet support vector machine prediction algorithm for avionics PHM," in *Intelligent Computing Theories*, pp. 295–304, Springer, 2013.
- [9] D. N. Thatoi, H. C. Das, and D. R. Parhi, "Review of techniques for fault diagnosis in damaged structure and engineering system," *Advances in Mechanical Engineering*, vol. 2012, Article ID 327569, 11 pages, 2012.
- [10] C. Bunks, D. McCarthy, and T. Al-Ani, "Condition-based maintenance of machines using hidden Markov models," *Mechanical Systems and Signal Processing*, vol. 14, no. 4, pp. 597–612, 2000.
- [11] P. Baruah and R. B. Chinnam, "HMMs for diagnostics and prognostics in machining processes," *International Journal of Production Research*, vol. 43, no. 6, pp. 1275–1293, 2005.
- [12] R. B. Chinnam and P. Baruah, "Autonomous diagnostics and prognostics through competitive learning driven HMM-based clustering," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 4, pp. 2466–2471, July 2003.
- [13] F. Camci and R. B. Chinnam, "Health-state estimation and prognostics in machining processes," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 581–597, 2010.
- [14] H. Ocak, K. A. Loparo, and F. M. Discenzo, "Online tracking of bearing wear using wavelet packet decomposition and probabilistic modeling: a method for bearing prognostics," *Journal of Sound and Vibration*, vol. 302, no. 4–5, pp. 951–961, 2007.
- [15] A. H. Tai, W.-K. Ching, and L. Y. Chan, "Detection of machine failure: hidden Markov Model approach," *Computers and Industrial Engineering*, vol. 57, no. 2, pp. 608–619, 2009.
- [16] Z.-J. Zhou, C.-H. Hu, D.-L. Xu, M.-Y. Chen, and D.-H. Zhou, "A model for real-time failure prognosis based on hidden Markov model and belief rule base," *European Journal of Operational Research*, vol. 207, no. 1, pp. 269–283, 2010.
- [17] D. A. Tobon-Mejia, K. Medjaher, N. Zerhouni, and G. Tripot, "A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models," *IEEE Transactions on Reliability*, vol. 61, no. 2, pp. 491–503, 2012.
- [18] Z. Liu, Q. Li, X. Liu, and C. Mu, "A hybrid LSSVR/HMM-based prognostic approach," *Sensors*, vol. 13, no. 5, pp. 5542–5560, 2013.
- [19] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [20] W. Li, Z. W. Wang, Z. C. Zhu, G. B. Zhou, and G. A. Chen, "Design of online monitoring and fault diagnosis system for belt conveyors based on wavelet packet decomposition and support Vector machine," *Advances in Mechanical Engineering*, vol. 2013, Article ID 797183, 10 pages, 2013.
- [21] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [22] M. Espinoza, J. A. K. Suykens, and B. De Moor, "Fixed-size least squares support vector machines: a large scale application in electrical load forecasting," *Computational Management Science*, vol. 3, no. 2, pp. 113–129, 2006.
- [23] C. Williams and M. Seeger, "Using the nystroem method to speed up Kernel machines," *Advances in Neural Information Processing Systems*, vol. 13, pp. 682–688, 2001.
- [24] A. Saxena and K. Goebel, "C-MAPSS Data Set," NASA Ames Prognostics Data Repository, NASA Ames, Moffett Field, Calif, USA, 2008, <http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/>.
- [25] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, July 2008.
- [26] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Metrics for offline evaluation of prognostic performance," *International Journal of Prognostics and Health Management*, vol. 1, no. 1, p. 20, 2010.
- [27] A. Saxena, J. Celaya, E. Balaban et al., "Metrics for evaluating performance of prognostic techniques," in *Proceedings of the International Conference on Prognostics and Health Management*, pp. 1–17, 2008.
- [28] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.