Building Local Models for Flexible Degradation Modeling and Prognostics

Changyue Song, *Member, IEEE*, Ziqian Zheng, and Kaibo Liu, *Senior Member, IEEE*

***Abstract*— To avoid unexpected failures of engineering systems, sensors have been widely used to monitor the degradation process of the systems. A number of studies have been conducted to analyze the collected sensor signals and predict the failure time. However, the existing studies are usually restricted and cannot be adapted to different practical situations. In this paper, we propose a systematic method for degradation modeling and prognosis that can be widely applied in different scenarios. In particular, the proposed method is capable to handle one or multiple sensors, powerful to capture the nonlinear relations between sensor signals and the degradation process with few assumptions, generic to consider multiple failure modes, flexible to deal with unequally spaced sensor measurements or asynchronous signals, and easily understandable with little preprocessing required. The main idea is to predict the failure time of an in-service unit based on a subset of the nearest historical units, where features are extracted from each sensor to describe the progression of sensor signals and local linear regression models are constructed to establish the relation between failure time and the extracted features. The prediction variance is then used as the goodness-of-fit measure, based on which decision-level fusion and feature-level fusion are proposed to combine multiple sensors. A case study with two datasets on the degradation modeling of aircraft engines is conducted which shows satisfactory performance of the proposed method.**

***Note to Practitioners*—This paper aims at modeling the collected sensor signals to understand the degradation process of the monitored engineering systems and predict the failure time. The main idea is to measure the similarity of units and predict the failure time of an in-service unit based on a subset of the nearest historical units. The developed method is widely applicable in different practical situations such as multiple sensors, multiple failure modes, asynchronous signals, and missing data. Furthermore, the method requires little preprocessing. There are several steps involved for implementing the proposed method: 1) collecting the sensor signals for historical units and the in-service unit; 2) extracting features from each sensor signal; 3) constructing a local linear model to predict the failure time based on the extracted features, and obtaining the prediction variance on the in-service unit; and 4) combining the information of different sensors using the decision-level fusion or feature-level fusion, if each unit is monitored by multiple sensors.**

***Index Terms*— Asynchronous signals, data fusion, failure time prediction, local linear regression, multiple failure modes.**

I. Introduction

DEGRADATION is inevitable in many engineering systems (also called units in this paper), such as aircraft engines, automotive batteries, and machining tools. Without proper maintenance, degradation may lead to unit failure, causing a variety of issues such as production downtime, logistic interruption, and safety issues. To avoid unexpected failures, sensors have been widely used to monitor the degradation status of a unit. By analyzing the collected sensor signals, the failure time of a unit can be predicted to provide guidance for maintenance [1]. Degradation modeling and prognosis based on the collected sensor signals are critical for successful predictive maintenance, and thus have attracted much research interest. Recent reviews on this topic can be found in [2]–[5].

Unfortunately, there are great challenges that have not been fully addressed in analyzing the sensor signals for degradation modeling and prognosis. These challenges are listed as follows.

(1) *How to handle multiple sensors*. Due to the rapid development of sensing technology, multiple sensors have been widely used to simultaneously monitor the same unit to acquire comprehensive information [6]. Since different sensor signals usually show different characteristics, they need to be effectively combined to produce a more accurate prediction of the failure time.

(2) *How to capture the complex and nonlinear relations between sensor signals and the degradation process.* Sensor measurements commonly do not directly reflect the degradation status or imply the failure time. Consequently, models are desired to establish the relation between sensor signals and the degradation process. However, the relation is usually unknown and can be complex and nonlinear.

(3) *How to consider multiple failure modes*. A unit may contain multiple components, and the failure may occur in any of the components, leading to multiple failure modes. Typically, the failure mode of a unit is unknown, and even the number of possible failure modes can be unknown in practice.

(4) *How to deal with unequally spaced sensor measurements and asynchronous signals*. In practice, due to sensor malfunctions or other reasons, the collected sensor

Manuscript received July 8, 2021; revised September 4, 2021; accepted October 26, 2021. This article was recommended for publication by Associate Editor B. Zhang and Editor F.-T. Cheng upon evaluation of the reviewers' comments. This work was supported in part by the Office of Naval Research under Grant N00014-17-1-2261 and in part by the Department of Energy under Award DE-NE0008805. *(Corresponding author: Changyue Song.)*

Changyue Song is with the School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: csong14@stevens.edu).

Ziqian Zheng and Kaibo Liu are with the Department of Industrial and Systems Engineering, University of Wisconsin–Madison, Madison, WI 53706 USA (e-mail: zzheng92@wisc.edu; kliu8@wisc.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2021.3124144.

Digital Object Identifier 10.1109/TASE.2021.3124144

measurements may not be equally spaced in time. In addition, when multiple sensors are used to monitor a single unit, different sensors may have different sampling rates, and thus result in asynchronous sensor signals. The asynchronous sensor signals lead to significant challenges for data fusion.

(5) *How to enhance the model interpretability to practitioners.* The model for analyzing the sensor signals should be easily understandable. If the model is too complicated and behaves like a black box, it will be too difficult for practitioners to interpret and implement the model confidently.

In the literature, parametric models such as general path models [7], [8] and stochastic processes [9]–[11] have been widely used. For example, Lu and Meeker proposed to use a mixed-effect model to account for the variation among different units [7]. However, parametric models are usually not flexible to consider multiple failure modes, especially when the number of possible failure modes is unknown. In addition, the assumptions of the parametric models may be restrictive and only hold in certain situations, and thus different models are required for different datasets. Also, the sensor signals often need to be preprocessed and transformed into a specific form [8] to satisfy the assumptions. However, in practice, there is often no guidance for the appropriate signal preprocessing or transformation. Recently, nonparametric models have also been proposed for degradation modeling. For example, Kontar *et al.* used multivariate Gaussian process to explore the interrelation of the signals from different units [12]. Chehade and Liu proposed to represent the collected signal from a unit of interest as a linear combination of other units [13], while Lin *et al.* tried to formulate a number of canonical models as a set of basis to represent the sensor signals of different units [14]. However, these models are complicated and require intensive computation. In addition, the methods mentioned above are limited to a single signal, and they mainly focus on predicting the future sensor readings instead of directly predicting the failure time. An additional step is usually required to obtain the predicted failure time, which may affect the prediction performance.

For units monitored by multiple sensors, health index approaches [15]–[17] and functional principal component analysis [18] have been proposed to fuse different sensor signals. Hidden Markov models can also be used to directly consider multiple sensors [19], [20]. However, these approaches are limited to synchronous signals. In addition, the methods are restricted to specific assumptions, and thus can only be used in certain situations. For example, most of the health index approaches only focus on fusing sensor signals by a linear function, but many applications may exhibit nonlinear functions with unknown functional forms. Recently, Song *et al.* proposed a new concept called failure surface to define unit failure based on multiple sensor signals to handle asynchronous signals [21], and Fang *et al.* extended [18] to highly incomplete signals by imputing the missing values [22], but the studies are limited to a single failure mode. To consider multiple failure modes, Chehade *et al.* proposed to linearly combine multiple signals into a single index to infer the unknown failure mode [23]. However, this method requires sensor signals to be synchronous and the failure modes of the training data set to be known ahead. Further, machine learning models such as artificial neural networks and support vector machines have also been proposed to directly predict the failure time based on the latest sensor observations [24]–[26]. These models enjoy the advantage of making less assumptions on the parametric forms of sensor signals or degradation process, but they cannot be easily interpreted and usually require the sensor signals to be synchronous or equally spaced in time.

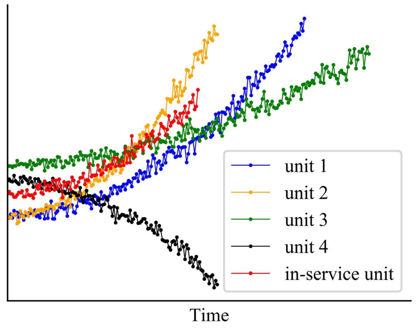


Fig. 1. Degradation signals for four historical units and one in-service unit.

To fill the literature gap, in this study, our objective is to *simultaneously* address *all* the aforementioned challenges and propose a systematic, widely applicable method for degradation modeling and prognosis based on one or multiple sensors. The proposed method is expected to: (1) be intuitive, easily understandable, and require little preprocessing of the sensor signals; (2) be able to capture the complex and nonlinear relations between sensor signals and the degradation process with few assumptions; (3) be generic to deal with multiple failure modes without knowing the possible number of failure modes or the ground true failure modes of the training data set; and (4) be flexible to deal with asynchronous and unequally-spaced sensor signals.

Our key idea is to predict the failure time of an in-service unit based on *a subset of the nearest historical units*, which is illustrated in Fig. 1. This figure shows the degradation signals for five units, where unit 1 to unit 4 are historical units that have already failed, and the remaining signal represents an in-service unit that has not failed yet. Intuitively, the failure time of the in-service unit is expected to be between that of unit 1 and unit 2, which are the nearest historical units. On the other side, knowing the failure time of unit 3 and unit 4 provides little information to help predict the failure time of the in-service unit, and thus they can be ignored. In particular, it is even necessary to ignore unit 4, which shows a decreasing trend in the signal in contrast to the increasing trend as in the in-service unit, indicating a different failure mode. Therefore, by building local models to only consider a subset of historical units, our method can be very flexible and generically consider multiple failure modes.

Indeed, the idea of using *nearest historical units* to predict the failure time of an in-service unit is not totally new. For example, an intuitive way is to measure the similarity of the in-service unit and the historical units, and then predict

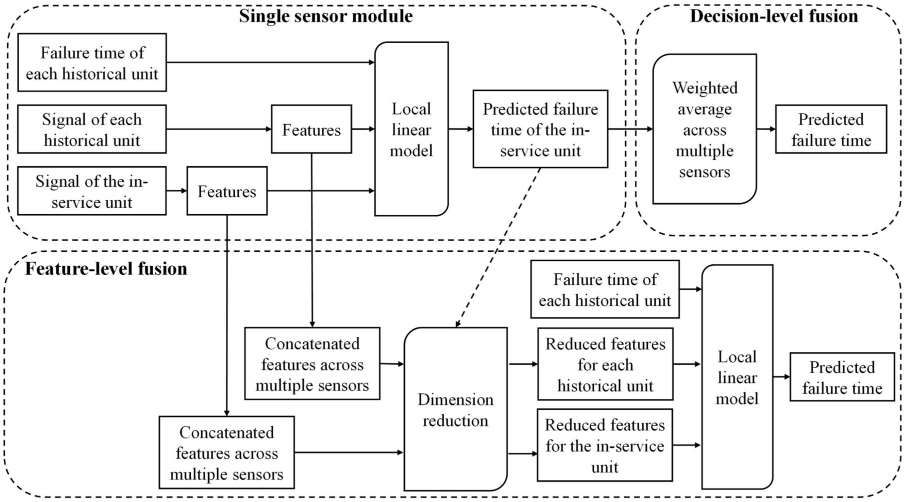

Single sensor module
Failure time of each historical unit
Signal of each historical unit
Signal of the in-service unit
Features
Features
Local linear model
Predicted failure time of the in-service unit
Decision-level fusion
Weighted average across multiple sensors
Predicted failure time
Feature-level fusion
Failure time of each historical unit
Concatenated features across multiple sensors
Concatenated features across multiple sensors
Dimension reduction
Reduced features for each historical unit
Reduced features for the in-service unit
Local linear model
Predicted failure time

Fig. 2. Flowchart of the proposed method.

the failure time of the in-service unit as a weighted average of the nearest historical units [27], [28]. However, metrics that measure unit similarity directly based on sensor signals such as the pointwise difference and Pearson correlation in the literature [27], [28] usually require the signals to be synchronous across different units and equally spaced in time. In addition, simply taking a weighted average of the historical units to predict the failure time of the in-service unit can only be used for interpolation but not for extrapolation. This is because the weights usually do not indicate whether a historical unit has a larger or smaller failure time than the in-service unit.

To address these problems, we propose a novel framework for flexible degradation modeling and prognosis as illustrated in Fig. 2. In particular, each sensor is first analyzed separately according to the single sensor module, and then different sensors are combined based on the decision-level fusion or feature-level fusion. For a single sensor, our innovative idea is to extract features from the signals of the in-service unit and each historical unit, and build local linear models to approximate the nonlinear relations between the features and the failure time within a small neighborhood of the in-service unit, which provides an easy-to-understand interpretation of the relation, and enjoys the extrapolation capability within the small neighborhood. In this way, a prediction on the failure time of the in-service unit is acquired based on a single sensor. Then, decision-level fusion can be employed to directly combine the predictions based on multiple sensors to produce a final prediction. Alternatively, feature-level fusion concatenates the features extracted from different sensors for each unit, and then builds a local linear model for failure time prediction after reducing the dimension of the features. As we will elaborate in Section III-B, the variance of the predicted failure time based on each single sensor can indicate whether the sensor is suitable for prognosis, and this information is then incorporated in the dimension reduction. It is worth noting that the proposed idea is possible to be extended to predict future sensor measurements as well, but in this paper, we do not explore this direction.

The rest of the paper is organized as follows. In Section II, we consider modeling and prognosis based on a single sensor, which is the single sensor module. In Section III, we discuss two strategies to combine multiple sensors. We conduct a case study with two datasets to validate our method in Section IV. Finally, Section V draws the conclusions and discusses future studies.

II. Prognosis With a Single Sensor

To begin with, in this section, we only consider a single sensor. Assume there is an in-service unit of interest, denoted as unit 0, and a set of historical units $\mathcal{M} = \{1, 2, \cdots, m\}$. Let $L_i(t)$ be the sensor measurement for unit i at time t where $i \in \{0\} \cup \mathcal{M}$, and T_i be the failure time of historical unit i where $i \in \mathcal{M}$. Our goal is to accurately predict the failure time of the in-service unit 0.

A. Feature Extraction

The first step is feature extraction. Without loss of generality, we decompose a signal as

$$L_i(t) = \eta_i(t) + \varepsilon_i(t), \quad i \in \{0\} \cup \mathcal{M}, \tag{1}$$

where $\eta_i(t)$ is the underlying true signal path, and $\varepsilon_i(t)$ is the noise and is not related to the failure time. For the in-service unit, the latest sensor measurements are more closely related to the failure time. Therefore, given τ to be the time when the latest measurement of the in-service unit is acquired, we propose to use $\left[\eta_i(\tau), \eta_i'(\tau), \eta_i''(\tau)\right]^T$ as the features for unit $i \in \{0\} \cup \mathcal{M}$, where $\eta_i'(t) = \partial\eta(t)/\partial t$ and $\eta_i''(t) = \partial^2\eta(t)/\partial t^2$. In other words, we use the zero-order, first-order, and second-order derivatives of $\eta_i(t)$ at time τ as the features for unit i. Here, the magnitude $\eta_i(\tau)$ of the signal is expected to be closely related to the current degradation status of the unit, and the first-order derivative $\eta_i'(\tau)$ represents the degradation rate and thus affects the future progression. The second-order derivative $\eta_i''(\tau)$ represents the curvature of the signal which indirectly affects $\eta_i(\tau)$ by affecting $\eta_i'(\tau)$. Higher order derivatives can also be used, but they are seldomly considered in practical degradation applications [8], and thus we do not employ higher order derivatives in this study. In fact, in some areas such as nonlinear optimization, it is a common practice to only consider up to the second-order derivative [29]. For historical unit $i \in \mathcal{M}$, we consider the derivatives at time τ as well to provide a fair comparison with the in-service unit. If a historical unit i failed before time τ, we regard $\eta_i(t)$ as a function of $t \in (0, \infty)$ and conceptually extend $\eta_i(t)$ beyond the failure time to calculate the features. In addition, since the features are directly based on $\eta_i(t)$ instead of the sensor signal $L_i(t)$, the method does not require the sensor measurements to be synchronous or equally spaced across all units. Note that based on the specific application, other features can also be extracted to characterize the progression of the signal.

To extract the features, at first, we need to estimate $\eta_i(t)$. In this study, we use the monotonic B-spline of degree 2 with infinite support to approximate $\eta_i(t)$ [30]. The advantage of B-spline is the flexibility to approximate complex functions. It also allows us to restrict $\eta_i(t)$ to be monotonic, which is desirable because degradation is known to be a monotonic process. Furthermore, once $\eta(t)$ is estimated, the expression of $\eta_i'(t)$ and $\eta_i''(t)$ can be easily derived as well. The degree of the B-spline is chosen to be 2, i.e., the basis functions are quadratic, because a higher degree may lead to overfitting, and according to the literature [30], [31], a degree of 2 is usually flexible enough in practice. If higher flexibility is desired, a degree of 3 can be chosen in a similar way.

In particular, the corresponding basis functions $\boldsymbol{\psi}_i(t) = \left[\psi_{i,0}(t), \psi_{i,1}(t), \cdots, \psi_{i,K+2}(t)\right]^T$ of the degree 2 B-spline can be derived using the induction formula (see Appendix for details), where K is the number of knots. In this study, we follow the literature [30] to use equally spaced knots, and K can be determined by cross validation or statistical information criteria such as AIC or BIC. In general, with restrictions on the monotonicity, the resulting splines are not sensitive to knot choices [32]. Then, we approximate $\eta_i(t)$ by

$$\eta_i(t) = \boldsymbol{\psi}_i(t)^T \boldsymbol{\Gamma}_i, \tag{2}$$

where $\boldsymbol{\Gamma}_i = \left[\Gamma_{i,0}, \Gamma_{i,1}, \cdots, \Gamma_{i,K+2}\right]^T$ can be estimated based on the collected sensor measurements. Specifically, let $\boldsymbol{L}_i = \left[L_i(t_1), \cdots, L_i\left(t_{n_i}\right)\right]^T$ be all the sensor measurements of unit i collected from time t_1 to t_{n_i}, and $\boldsymbol{\Psi}_i = \left[\boldsymbol{\psi}_i(t_1)^T; \cdots; \boldsymbol{\psi}_i\left(t_{n_i}\right)^T\right]$ be the corresponding design matrix. Since

$$\boldsymbol{L}_i = \boldsymbol{\Psi}_i \boldsymbol{\Gamma}_i + \boldsymbol{\varepsilon}_i, \tag{3}$$

we can estimate $\boldsymbol{\Gamma}_i$ by minimizing $(\boldsymbol{L}_i - \boldsymbol{\Psi}_i\boldsymbol{\Gamma}_i)^T(\boldsymbol{L}_i - \boldsymbol{\Psi}_i\boldsymbol{\Gamma}_i)$. According to [30], by imposing proper constraints, the monotonicity of $\eta_i(t)$ can be ensured. For example, the following constraints ensure $\eta_i(t)$ to be increasing:

$$\begin{aligned} \Gamma_{i,k} &\le 0, \quad k = 0, 1, \\ \Gamma_{i,k} - \Gamma_{i,k-1} &\ge 0, \quad k = 3, \cdots, K, \\ \Gamma_{i,k} &\ge 0, \quad k = K+1,\ K+2. \end{aligned} \tag{4}$$

If we denote $\hat{\boldsymbol{\Gamma}}_i$ as the estimation, then the features are extracted as $\eta_i(\tau) = \boldsymbol{\psi}_i(\tau)^T\hat{\boldsymbol{\Gamma}}_i$, $\eta_i'(\tau) = \boldsymbol{\psi}_i'(\tau)^T\hat{\boldsymbol{\Gamma}}_i$, and $\eta_i''(\tau) = \boldsymbol{\psi}_i''(\tau)^T\hat{\boldsymbol{\Gamma}}_i$, where $\boldsymbol{\psi}_i'(t)$ and $\boldsymbol{\psi}_i''(t)$ are the first-order and second-order derivatives of the basis functions, which can be obtained by the induction formula. Please refer to the Appendix for details. The extracted features are then standardized, and we use $\boldsymbol{x}_i$ to denote the standardized features, i.e.,

$$x_i^{(p)} = \frac{\eta_i^{(0)}(\tau) - \mu_\eta^{(p)}}{\sigma_\eta^{(p)}}, \tag{5}$$

where $x_i^{(p)}$ is the pth entry of $\boldsymbol{x}_i$ $(p = 0, 1, 2)$, $\eta_i^{(0)}(\tau), \eta_i^{(1)}(\tau), \eta_i^{(2)}(\tau)$ represents the original features $\eta_i(\tau), \eta_i'(\tau), \eta_i''(\tau)$, respectively, and $\mu_\eta^{(p)}, \sigma_\eta^{(p)}$ are the average and standard deviation of $\left\{\eta_i^{(p)}(\tau), \forall i \in \mathcal{M}\right\}$, $p = 0, 1, 2$.

B. Local Linear Regression

After feature extraction, we establish the relation between the failure time T_i and the extracted features $\boldsymbol{x}_i$. Although the relation $T = f(\boldsymbol{x}) + \varepsilon$ can be nonlinear and the functional form is unknown, we do not need to estimate the entire relation $f(\cdot)$. Instead, we are only interested in a small neighborhood $\mathcal{N}(\boldsymbol{x}_0)$ of the in-service unit $\boldsymbol{x}_0$. Specifically, we have

$$T = f(\boldsymbol{x}) + \varepsilon = f(\boldsymbol{x}_0) + \left[f'(\boldsymbol{x}_0)\right]^T(\boldsymbol{x} - \boldsymbol{x}_0) + C(\boldsymbol{x}) + \varepsilon,$$

where $f'(\boldsymbol{x}_0)$ is the gradient of $f(\boldsymbol{x})$ at $\boldsymbol{x}_0$, and the remainder $C(\boldsymbol{x}) \to 0$ as $\boldsymbol{x} \to \boldsymbol{x}_0$. Within a small neighborhood $\boldsymbol{x} \in N(\boldsymbol{x}_0)$, we can use a linear model to approximate the nonlinear relation

$$T \approx f(\boldsymbol{x}_0) + (\boldsymbol{x} - \boldsymbol{x}_0)^T f'(\boldsymbol{x}_0) + \varepsilon = \left[1, \boldsymbol{x}^T\right]\boldsymbol{\beta}_0 + \varepsilon,$$

where $\boldsymbol{\beta}_0 = \left[f(\boldsymbol{x}_0) - \boldsymbol{x}_0^T f'(\boldsymbol{x}_0); f'(\boldsymbol{x}_0)\right]$, and $\varepsilon \sim N(0, \sigma_0)$ is assumed to be the white Gaussian noise. This motivates us to consider local linear regression (LLR) based on the nearest historical units. Our idea is illustrated in Fig. 3(a), where each dot denotes the extracted feature and the failure time of one historical unit, the black dashed curve represents the unknown function $f(\cdot)$, the vertical line indicates the extracted feature of the in-service unit, and the blue dots are the nearest historical units. The coefficient $\boldsymbol{\beta}_0$ provides an intuitive interpretation on the relation between features and the failure time. For example, a negative entry of $\boldsymbol{\beta}_0$ indicates that the corresponding feature of $\boldsymbol{x}$ is negatively related with the failure time T, and thus a unit with a greater value of this feature is expected to have a smaller failure time. It is worth noting that the coefficient $\boldsymbol{\beta}_0$ is unique for the in-service unit, and different in-service units will have different coefficient $\boldsymbol{\beta}_0$s.

To estimate $\boldsymbol{\beta}_0$, at first, we select a kernel function

$$\mathcal{K}(c_i) = \mathcal{K}\left(\|\boldsymbol{x}_0 - \boldsymbol{x}_i\|^2\right)$$

to assign a weight $\mathcal{K}(c_i) \ge 0$ to each historical unit i, where $c_i = \|\boldsymbol{x}_0 - \boldsymbol{x}_i\|^2$ measures the Euclidean distance between historical unit i and the in-service unit. Historical units that are close to the in-service unit are assigned with high weights, while other historical units are assigned with low or even zero weights. For example, the k-nearest-neighbor kernel function can be defined as:

$$\mathcal{K}(c_i) = \begin{cases} 1, & c_i \le c^{(n)} \\ 0, & c_i > c^{(n)} \end{cases} \tag{6}$$

where $c^{(n)}$ is the nth smallest value of $\{c_i, i \in \mathcal{M}\}$. In other words, the k-nearest-neighbor kernel relies only on the n nearest historical units to build a linear model, where each of the n nearest historical units is regarded as equally important, and all the other historical units are discarded.

After the weights $\{\mathcal{K}(c_i), i \in \mathcal{M}\}$ are computed, $\boldsymbol{\beta}_0$ is estimated by minimizing the weighted sum of square

$$\begin{aligned} h(\boldsymbol{\beta}_0) &= \sum_{i \in \mathcal{M}} \mathcal{K}(c_i)(T_i - [1, \boldsymbol{x}_i^T]\boldsymbol{\beta}_0)^2 \\ &= (\boldsymbol{T} - \boldsymbol{X}\boldsymbol{\beta}_0)^T \boldsymbol{K} (\boldsymbol{T} - \boldsymbol{X}\boldsymbol{\beta}_0) \end{aligned} \tag{7}$$

where $\boldsymbol{X} = \left[1, \boldsymbol{x}_i^T\right]_{i \in \mathcal{M}}$ is the m-by-4 design matrix, $\boldsymbol{T} = [T_i]_{i \in \mathcal{M}}$ is the m-by-1 response vector, and $\boldsymbol{K} = \mathrm{diag}(\mathcal{K}(c_1), \cdots, \mathcal{K}(c_m))$ is the diagonal weight matrix. The estimation is

$$\hat{\boldsymbol{\beta}}_0 = \{\boldsymbol{X}^T \boldsymbol{K} \boldsymbol{X}\}^{-1} \boldsymbol{X}^T \boldsymbol{K} \boldsymbol{T}. \tag{8}$$

Accordingly, we can easily obtain the predicted failure time for the in-service unit as

$$\hat{T}_0 = \left[1, \boldsymbol{x}_0^T\right] \hat{\boldsymbol{\beta}}_0.$$

With LLR, our method does not impose restrictive assumptions on the specific functional form of $f(\cdot)$. It also allows us to easily deal with multiple failure modes, because historical units with different failure modes will be far from the in-service unit and thus be discarded or assigned with a small weight in the LLR model. In contrast, many existing studies that focus on predicting the future sensor readings make assumptions on the functional form of the signal, and assume the unit fails when the signal crosses a pre-specified failure threshold [7]–[14]. However, in practice, these assumptions may not hold, and the failure threshold may be difficult to specify in cases such as multiple failure modes.

One issue here is how to measure the goodness-of-fit for the linear model. For the local linear model to be adequate, there are two requirements. First, the nearest historical units should show linear relation. In other words, $h\left(\hat{\boldsymbol{\beta}}_0\right)$ should be small. For example, as shown in Fig. 3(b), if too many historical units are involved in the local linear model, and the actual relation $f(\cdot)$ is highly nonlinear, the estimated model may be biased. Second, the nearest historical units should be close to the in-service unit. It is possible that all historical units in $\mathcal{M}$ are indeed far from the in-service unit, which is illustrated in Fig. 3(c). In this case, the nearest historical units are not within a small neighborhood of $\boldsymbol{x}_0$, and thus the linear approximation will not be appropriate. Therefore, to address this issue, we propose to use the variance of $\hat{T}_0$, denoted as υ, to measure the goodness of fit, which can be estimated as,

$$\hat{\upsilon} = Var\left(T_0 - \hat{T}_0\right) = \left[1, \boldsymbol{x}_0^T\right] \boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}_0} \begin{bmatrix} 1 \\ \boldsymbol{x}_0 \end{bmatrix} + \hat{\sigma}_0^2, \tag{9}$$

where the covariance matrix of $\hat{\boldsymbol{\beta}}_0$ is

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}_0} = \hat{\sigma}_0^2 \left\{\boldsymbol{X}^T \boldsymbol{K} \boldsymbol{X}\right\}^{-1} \boldsymbol{X}^T \boldsymbol{K}^2 \boldsymbol{X} \left\{\boldsymbol{X}^T \boldsymbol{K} \boldsymbol{X}\right\}^{-1}, \tag{10}$$

and

$$\hat{\sigma}_0^2 = \frac{h\left(\hat{\boldsymbol{\beta}}_0\right)}{tr(\boldsymbol{K}) - tr\left[\left(\boldsymbol{X}^T \boldsymbol{K} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{K}^2 \boldsymbol{X}\right]}. \tag{11}$$

See Chapter 2 and Chapter 9 of [33] for the detailed derivation. The prediction variance $\hat{\upsilon}$ will be large if the nearest historical units do not show linear relation, or if the in-service unit is far from the historical units, as in the cases of Fig. 3(b) and Fig. 3(c). In addition, $\hat{\upsilon}$ provides a prediction interval for $\hat{T}_0$, which is $\left[\hat{T}_0 - \Phi^{-1}(1 - \alpha/2) \cdot \sqrt{\hat{\upsilon}}, \hat{T}_0 + \Phi^{-1}(1 - \alpha/2) \cdot \sqrt{\hat{\upsilon}}\right]$ with a confidence level of $1 - \alpha$, where $\Phi^{-1}(1 - \alpha/2)$ is the $1 - \alpha/2$ quantile of the standard normal distribution. A small $\hat{\upsilon}$ indicates a narrower confidence interval and thus less uncertainty of $\hat{T}_0$.

III. Prognosis With Multiple Sensors

In this section, we consider the case when each unit $i \in \{0\} \cup \mathcal{M}$ is monitored by a set of sensors $\mathcal{S} = \{1, 2, \cdots, s\}$. Specifically, we apply the method in Section II separately to each sensor. Denote $\boldsymbol{x}_{i,j}$ as the extracted features for unit i from sensor j, $\hat{T}_{0,j}$ as the predicted failure time for the in-service unit based on sensor j, and $\hat{\upsilon}_j$ as the corresponding prediction variance for sensor j. We propose two strategies, including decision-level fusion and feature-level fusion [34] to combine the sensors.

A. Decision-Level Fusion

The most straightforward way of combining sensors is to directly combine the predicted failure time from different sensors. However, one common challenge is how to determine the weight of each sensor during the fusion. In this study, a reasonable choice is to consider the weight of sensor j based on the prediction variance υ_j. In particular, we propose to consider the final predicted failure time for the in-service unit by

$$\hat{T}_0 = \frac{\sum_{j \in \mathcal{S}} w_j \hat{T}_{0,j}}{\sum_{j \in \mathcal{S}} w_j}. \tag{12}$$

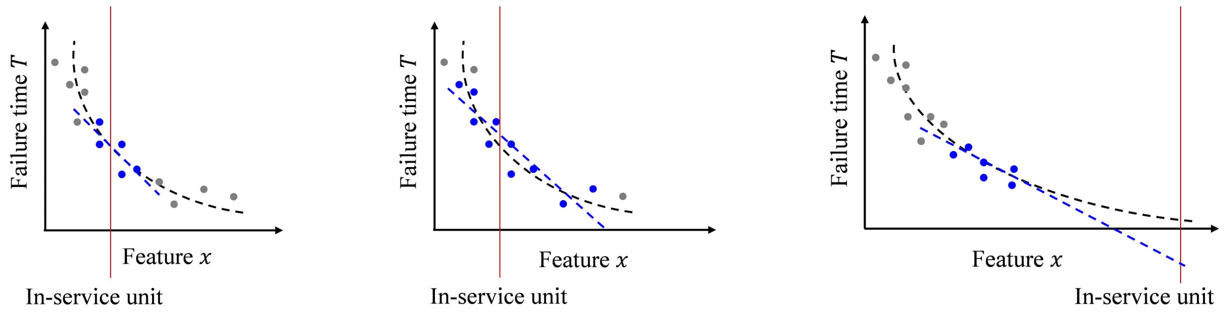
Here,

$$w_j = \begin{cases} \left(\dfrac{1}{\hat{\upsilon}_j}\right)^{\gamma}, & \hat{\upsilon}_j \le \hat{\upsilon}^{(q)} \\ 0, & \hat{\upsilon}_j > \hat{\upsilon}^{(q)} \end{cases} \tag{13}$$

is the weight for $\hat{T}_{0,j}$ based on the prediction variance $\hat{\upsilon}_j$, and $\hat{\upsilon}^{(q)}$ is the qth smallest value of $\left\{\hat{\upsilon}_j, j \in \mathcal{S}\right\}$. The prediction variance $\hat{\upsilon}_j$ is adjusted by a parameter $\gamma \ge 0$. The rationale is that a sensor with smaller prediction variance υ_j should get a larger weight w_j. In addition, according to (13), only the q sensors with the smallest goodness-of-fit measure $\hat{\upsilon}_j$ have a nonzero weight, and other sensors are discarded. The selection of q and γ will be discussed in Section IV-E.

B. Feature-Level Fusion

Another way of combining the sensors is to combine the extracted features. Denote τ_j as the time when the latest observation of the in-service unit from sensor j is collected. We extract features $\left[\eta_{i,j}(\tau_j), \eta'_{i,j}(\tau_j), \eta''_{i,j}(\tau_j)\right]^T$ from each sensor j for each unit $i \in \{0\} \cup \mathcal{M}$ as in Section II.A, which are denoted as $\boldsymbol{x}_{i,j} = \left[x_{i,j}^{(0)}, x_{i,j}^{(1)}, x_{i,j}^{(2)}\right]^T$ after standardization



(a) (b) (c)

Fig. 3. Illustration of the local linear model based on the nearest historical units, where (a) shows the ideal case, (b) shows the case when the local model is not linear with too many historical units included, and (c) shows the case when the in-service unit is too far away from the historical units.

according to (5). Let $\boldsymbol{x}_i = \left[\boldsymbol{x}_{i,j}\right]_{j\in\mathcal{S}}$ be the $3s$-by-1 concatenated features for unit $i \in \{0\} \cup \mathcal{M}$. Since the dimension of $\boldsymbol{x}_i$ can be large, directly building an LLR to predict the failure time T_i may lead to overfitting. Therefore, dimension reduction is considered by $z = \boldsymbol{\phi}(\boldsymbol{x})$, where a mapping $\boldsymbol{\phi}(\cdot)$ is used to fuse the feature $\boldsymbol{x}$ into feature z with reduced dimension. The challenge here is that the mapping $\boldsymbol{\phi}(\cdot)$ is unknown and may be nonlinear. To address the challenge, we again build local models to approximate $\boldsymbol{\phi}(\cdot)$ within a small neighborhood of the in-service unit $\boldsymbol{x}_0$. The idea is indeed similar as in LLR of Section II-B. Specifically, at point $\boldsymbol{x}_0$, we can expand

$$\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x}_0) + \left[\boldsymbol{\phi}'(\boldsymbol{x}_0)\right]^T (\boldsymbol{x} - \boldsymbol{x}_0) + \boldsymbol{C}(\boldsymbol{x}),$$

where $\boldsymbol{\phi}'(\boldsymbol{x})$ is the gradient matrix of $\boldsymbol{\phi}(\boldsymbol{x})$, and $\boldsymbol{C}(\boldsymbol{x})$ goes to $\boldsymbol{0}$ as $\boldsymbol{x} \to \boldsymbol{x}_0$. Therefore, within a small neighborhood of $\boldsymbol{x}_0$, we can well approximate

$$\begin{aligned} z = \boldsymbol{\phi}(\boldsymbol{x}) &\approx \boldsymbol{\phi}(\boldsymbol{x}_0) + \left[\boldsymbol{\phi}'(\boldsymbol{x}_0)\right]^T (\boldsymbol{x} - \boldsymbol{x}_0) \\ &= \boldsymbol{B}\boldsymbol{x} + \boldsymbol{A}, \end{aligned} \tag{14}$$

where the gradient matrix $\boldsymbol{B} = \left[\boldsymbol{\phi}'(\boldsymbol{x}_0)\right]^T$ is the key to fuse the features, and $\boldsymbol{A} = \boldsymbol{\phi}(\boldsymbol{x}_0) - \boldsymbol{B}\boldsymbol{x}_0$ adds a constant to all units and thus can be ignored. With the gradient $\left[\boldsymbol{\phi}'(\boldsymbol{x}_0)\right]^T$, we are essentially approximating the nonlinear mapping $\boldsymbol{\phi}$ for fusing the features locally by its tangent plane at $\boldsymbol{x}_0$.

To derive $\left[\boldsymbol{\phi}'(\boldsymbol{x}_0)\right]^T$ for fusing the features, we apply the principal component analysis (PCA) to the nearest historical units, where the overall distance between unit i and the in-service unit is defined as

$$\delta_i = \sum_{j\in\mathcal{S}_\upsilon(q)} \frac{\left\|\boldsymbol{x}_{i,j} - \boldsymbol{x}_{0,j}\right\|^2}{\hat{\upsilon}_j}$$

Here $\mathcal{S}_\upsilon(q) = \left\{j \in \mathcal{S} : \hat{\upsilon}_j \le \hat{\upsilon}^{(q)}\right\}$ is the set of q sensors with the smallest value of prediction variance $\hat{\upsilon}_j$, and $\hat{\upsilon}_j$ is used to adjust the contribution of sensor j in determining the distance δ_i. In other words, sensors with smaller υ_j will be given higher weights when calculating δ_i. Then, we can identify the closest n' historical units as

$$\mathcal{M}_\delta(n') = \left\{i \in \mathcal{M} : \delta_i \le \delta^{(n')}\right\}, \quad n' \le m,$$

where $\delta^{(n')}$ is the n'th smallest value of $\{\delta_i, i \in \mathcal{M}\}$.

In this study, since the features extracted from each signal have different meanings, we apply PCA to each dimension of features separately. In particular, we denote $\boldsymbol{X}^{(p)} = \left[x_{i,j}^{(p)}\right]_{i\in\mathcal{M}_\delta(n'), j\in\mathcal{S}_\upsilon(q)} \in \mathcal{R}^{n'\times q}$, $p = 0, 1, 2$ as the collection of the zero-order, first-order, and second-order derivatives for the historical units in $\mathcal{M}_\delta(n')$ and sensors in $\mathcal{S}_\upsilon(q)$. PCA is then applied to each $\boldsymbol{X}^{(p)}$, and let $\rho_j^{(p)}$ be the loadings of the first principal component. The fused features for each unit $i \in \{0\} \cup \mathcal{M}$ is $z_i = \left[z_i^{(0)}, z_i^{(1)}, z_i^{(2)}\right] \in \mathcal{R}^{1\times 3}$, where

$$z_i^{(p)} = \sum_{j\in\mathcal{S}_\upsilon(q)} \rho_j^{(p)} x_{i,j}^{(p)}.$$

It is worth noting that $z_i^{(p)}$ can be regarded as a projection of $\left\{x_{i,j}^{(p)}, \forall j \in \mathcal{S}_\upsilon(q)\right\}$ onto a one-dimensional space with maximum variance among the nearest historical units. We use $\{z_i, \forall i \in \mathcal{M}\}$ as the predictors and $\{T_i, \forall i \in \mathcal{M}\}$ as responses to build a local linear model, and then the failure time $\hat{T}_0$ can be predicted as in Section II.B.

IV. Case Study

In this section, we implement and evaluate our method using two datasets that involve the degradation of aircraft engines.

A. Data Description

The data are provided in [35], which were generated by C-MAPSS, a widely used model for simulating the degradation process of large commercial turbofan engines. There are four datasets. Each dataset consists of a number of historical units and in-service units. Depending on the dataset, the units are subject to one or multiple failure modes, and operate under one or multiple operational conditions. Table I summarizes the four datasets. In this study, we focus on Dataset 1 and Dataset 3, and the other two datasets with multiple operational conditions will be considered in the future study.

Each unit of the datasets is monitored by 21 sensors, measuring comprehensive information such as temperature and pressure. The detailed description of these sensors is given in Table II. Sensor measurements were simultaneously and continuously collected from the historical units until failure, while the signals were truncated before failure for in-service units. The actual remaining useful life (RUL) of each

TABLE I
SUMMARY OF THE FOUR DATASETS

Dataset	Historical Units	In-service Units	Failure Modes	Operational Conditions
1	100	100	1	1
2	260	259	1	6
3	100	100	2	1
4	248	249	2	6

TABLE II
DETAILED DESCRIPTION OF THE 21 SENSORS

Symbol	Description	Units
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/P2)	--
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	--
farB	Burner fuel-air ratio	--
htBleed	Bleed Enthalpy	--
Nf_dmd	Demanded fan speed	rpm
PCNfR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

in-service unit is recorded in a separate file. The task is to predict the failure time of the in-service units and compare with the ground truth.

B. Dataset 1

Dataset 1 contains 100 historical units and 100 in-service units under a single failure mode and a single operational condition. At first, we exclude sensors that only show constant measurements, resulting in 14 sensors remained. Then, we directly apply our proposed method for each in-service unit as illustrated in Fig. 2, as no further preprocessing is required. In particular, the single sensor module is used to analyze each sensor separately as described in Section II. And then decision-level fusion and feature-level fusion described in Section III are used to combine different sensors.

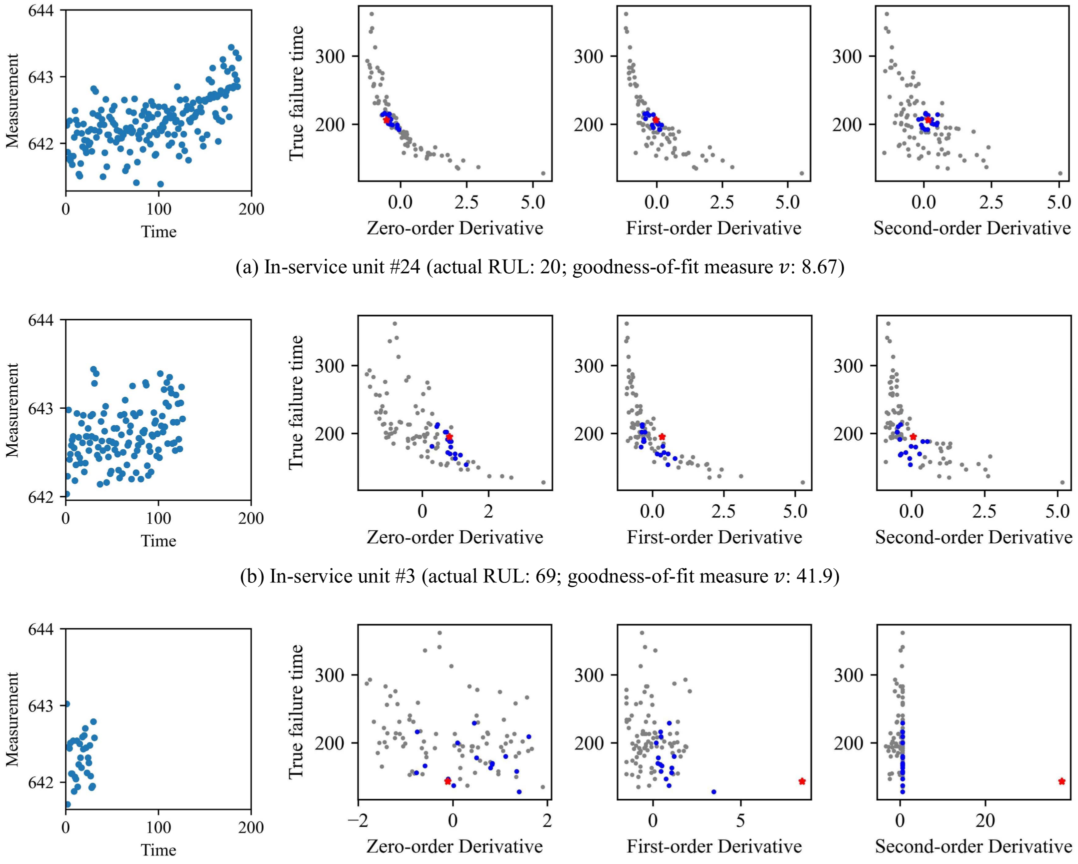
When analyzing each single sensor signal, we use the degree 2 monotonic B-spline to fit the signal and extract the features, where the B-splines knots are equally spaced and the number of knots is determined by BIC for each sensor and each unit. As a demonstration, we first consider the k-nearest-neighbor kernel function as in (6) with $n = 15$ to discuss the characteristics of the single sensor module. Later, we will conduct cross validation to select the best kernel function as described in Section IV.E.

To easily understand and validate the single sensor module of our method, we visualize the extracted features from the sensor T24 for three selected in-service units as shown in Fig. 4. For each in-service unit, the first plot shows the available sensor measurements, and the next three plots show the extracted three features after standardization as the x-axis, and the true failure time as the y-axis. For the three plots, the red points represent the in-service unit, the blue points represent the 15 historical units that are identified as the nearest neighbors, and the grey points represent the other historical units. It is also worth noting that the true failure time of each in-service unit is only used in Fig. 4 for illustration, and is not used in model construction or prognosis. As described in Section II.A, even for the same historical units, the extracted features may be different when analyzing different in-service units, as they depend on the time when the latest observation for the in-service units is collected.

Fig. 4(a) shows the in-service unit #24. If we consider all the blue and grey points, it can be observed that overall, failure time has a nonlinear relation $f(\cdot)$ with the extracted features. The zero-order derivative feature $\eta_i(\tau)$ seems to be more closely related to the failure time with denser blue and grey points than the other two features. This means that for in-service unit #24, the zero-order derivative feature is a stronger indicator of failure time. Fig. 4(b) shows the in-service unit #3, where the zero-order derivative is weaker than the other features, especially when the value of the feature is small. The reason is that the actual RUL of the in-service unit #24 is only 20, meaning that it is very close to failure. Therefore, the magnitude of the signal can largely indicate the degradation status of the unit. In contrast, the actual RUL of the in-service unit #3 is 69. Since the degradation level of the unit is still low, the magnitude of the sensor signal is not as sensitive as the slope, i.e., the rate of degradation, to indicate the failure time. For in-service unit #24 and unit #3, by only considering 15 nearest neighbors, we can well approximate the nonlinear relation $f(\cdot)$ locally with a linear model around the in-service unit, and our proposed goodness-of-fit measure υ is estimated as 8.67 and 41.9, respectively, indicating relatively good fit.

In Fig. 4(c), we show the in-service unit #1. The actual RUL is 112, meaning that the degradation of the in-service unit #1 is still at an early stage, and the sensor signal does not show a clear degradation trend yet. In addition, there are few sensor measurements available to fit the signal path $\eta_i(t)$ and extract the features. Consequently, the extracted features contain high uncertainty, and the nearest 15 neighbors are indeed far away from the in-service unit. In this case, it is not proper to build an LLR model for failure time prediction. The goodness-of-fit measure for unit #1 is 5.7×10^{12}, indicating a poor fit. This result illustrates that our proposed goodness-of-fit measure is effective in identifying inappropriate sensors.

In Table III, we show the estimated prediction variance $\hat{\upsilon}_j$ for unit #24, unit #3, and unit #1, based on each sensor. Generally, the prediction variances for unit #24 are small, while those for unit #1 are large. This agrees with our intuition because the trend of the signals for unit #24 is much clearer than unit #1. Another observation is that for different in-service units, our proposed method is flexible to identify different sets of sensors to use and assign different weights when combining them for prognosis. For example,



(a) In-service unit #24 (actual RUL: 20; goodness-of-fit measure v: 8.67)

(b) In-service unit #3 (actual RUL: 69; goodness-of-fit measure v: 41.9)

(c) In-service unit #1 (actual RUL: 112; goodness-of-fit measure v: 5.7×10^{12})

Fig. 4. Illustration of the extracted features from sensor T24 for three in-service units of Dataset 1. For each in-service unit, the first plot shows the available sensor measurements, and the next three plots show the true failure time versus three extracted features for the in-service unit (red point) and the historical units (blue points and grey points), where the blue points denote the 15 nearest neighbors of the in-service unit.

the prediction variance of T24 is satisfactory for unit #24 and unit #3, but extremely large for unit #1. Therefore, for unit #1, other sensors that are more reliable should be used for prognosis instead of T24. This is different from existing studies such as the health index approaches [15]–[17] which use fixed sensor sets and assign fixed weights to the sensors for all in-service units.

As mentioned before, Table III is obtained based on the k-nearest-neighbor kernel function with $n = 15$ for demonstration purpose. In this case study, we use cross validation to select the best kernel function for each sensor of each in-service unit. Three kernel functions are considered in this case study, including the k-nearest-neighbor kernel defined in (6), the triangular kernel defined as

$$\mathcal{K}(c_i) = \begin{cases} 1 - c_i/c^{(n)}, & c_i \le c^{(n)} \\ 0, & c_i > c^{(n)}, \end{cases}$$

and the parabolic kernel defined as

$$\mathcal{K}(c_i) = \begin{cases} 1 - \left(c_i/c^{(n)}\right)^2, & c_i \le c^{(n)} \\ 0, & c_i > c^{(n)}. \end{cases}$$

Then, we combine the multiple sensors using decision-level fusion and feature-level fusion, where the parameters involved are selected based on the approach described in Section IV.E.

TABLE III

PREDICTION VARIANCE FOR THREE IN-SERVICE UNITS BASED ON EACH SENSOR

Sensor	Unit #24	Unit #3	Unit #1
T24	8.67	41.9	5.7×10^{12}
T30	8.68	58.8	1.3×10^4
T50	45.0	14.8	8.3×10^4
P30	72.3	205	2188
Nf	412.7	131	1.2×10^7
Nc	18.4	1984	3128
Ps30	398.7	9.5	2867
phi	204.3	11.3	1236
NRf	523.0	145	5923
NRc	31.2	2605	2607
BPR	114.8	598	2.5×10^4
htBleed	38.8	53.5	8.7×10^4
W31	22.0	33.8	1.4×10^4
W32	19.2	368	1038

Moreover, we consider the failure surface method proposed by [21] as the benchmark method, because to the best of our knowledge, this method achieves one of the best performances for analyzing Dataset 1 in the literature. The failure surface method models each sensor signal parametrically, and trains a classifier to infer the degradation status of the in-service unit based on the fitted signal paths of all the sensors.

However, the failure surface method is limited to a single failure mode.

For an in-service unit 0, the failure time prediction error is defined as

$$\text{Error} = \frac{\left|\hat{T}_0 - T_0\right|}{T_0},$$

where T_0 is the true failure time, and $\hat{T}_0$ is the predicted failure time. The averaged failure time prediction errors for the in-service units of Dataset 1 using our proposed method with decision-level fusion and feature-level fusion are shown in Fig. 5. The x-axis of Fig. 5 means the levels of the actual RUL. For example, "80" means only the in-service units with actual RUL less than or equal to 80 are considered to calculate the average failure time prediction error. From Fig. 5, the prediction error of all methods decreases when the actual RUL decreases. This is because with a small actual RUL, the trends of the sensor signals become clearer and the prediction horizon becomes smaller. When the actual RUL is large, the failure surface method performs much better than the proposed method. This is because the failure surface method is parametric, and assumes all units to share some commonalities. Consequently, when the assumption is satisfied, the failure surface method is able to extract the commonalities from all the historical units as the prior information to compensate for the sparse available measurements of the in-service unit. However, if the assumption is not satisfied, the failure surface method will fail to produce a satisfactory prediction, which is the reason why it cannot deal with multiple failure modes. In contrast, to acquire flexibility, our proposed method is nonparametric with few assumptions, and we rely solely on the sensor measurements of each unit for feature extraction. When the actual RUL is large, the extracted features of the in-service unit still contain high uncertainty as the degradation trend is not clear yet. Consequently, our proposed method does not perform well in this case. However, as the actual RUL decreases, the performance of our proposed method quickly improves. When the actual RUL is less than or equal to 80, the feature-level fusion method achieves similar performance as the failure surface method, and the performance of the decision-level fusion method is even better. This result shows the superiority of our method for the case when actual RUL is not large and the features can be reliably extracted.

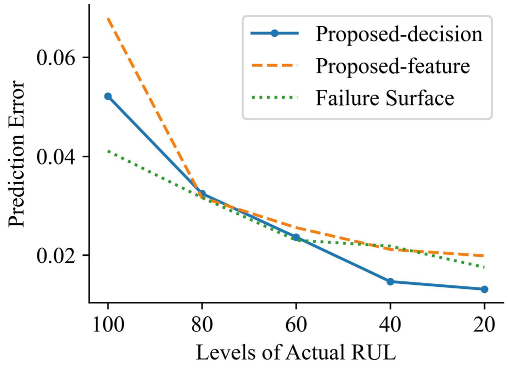


Fig. 5. Failure time prediction error for the in-service units of Dataset 1 using the failure surface benchmark method and the proposed method with the decision-level fusion and feature-level fusion.

C. Dataset 3

Dataset 3 contains 100 historical units and 100 in-service units, and each unit is subjected to two possible failure modes. The true failure mode of each historical unit is unknown, which makes the prognostic task more challenging. Although Dataset 3 is more complicated than Dataset 1, our proposed method can still be directly used to analyze Dataset 3 with little preprocessing.

Similar to Fig. 4, Fig. 6 shows the available measurements and extracted features from the sensor BPR for an in-service unit #94 with actual RUL to be 10. Clearly, the units can be divided into two groups, corresponding to two different failure modes. The in-service unit #94 belongs to the group where BPR is decreasing with a negative first-order derivative feature $\eta_0'(\tau) < 0$. However, as the figure shows, there are also units with increasing BPR signal (e.g., units with positive first-order derivative). In this case, without knowing the number of possible failure modes and the failure mode of each historical unit, it is difficult to identify proper failure thresholds for many existing models [7]–[14], which assume a unit fails when the signal crosses a failure threshold. In contrast, our method automatically identifies the failure mode of the in-service unit #94 based on the nearest neighbors. Since the actual RUL of the in-service unit #94 is small, the extracted features are highly reliable and show close relation with the true failure time. This example further shows that our method can be easily understood and visually evaluated, which is an important advantage in practice.

Since the failure surface method in Section IV.B is limited to one failure mode, it cannot be used to analyze Dataset 3. Instead, we consider the failure mode index method [23] as the benchmark, since it is designed specially to deal with the multiple failure modes challenge in Dataset 3. The idea of the failure mode index method is to combine the multiple sensor signals into a one-dimensional index, which can reflect the failure mode of the unit. Note that the failure mode index method is more restrictive than our proposed method because it requires the sensors to be synchronous and the failure modes of historical units to be known. Since the true failure mode of each historical unit is unknown, the authors in [23] have to heuristically compare Dataset 3 with Dataset 1 based on the sensor trends to pre-determine the failure mode of each historical unit.

Fig. 7 shows the averaged failure time prediction error using our method and compares with the failure mode index method. Our proposed method achieves much better performance, in addition to more flexibility. For Dataset 3, the performance of decision-level fusion is much better than the feature-level fusion, especially when the actual RUL is medium or large. One possible reason is that with a greater actual RUL, the extracted features $\boldsymbol{x}_i$ involves higher level of noise that is not related to the failure time. In feature-level fusion, when applying local PCA to the features $\boldsymbol{X}^{(p)}$, $p = 0, 1, 2$, an implicit assumption is that the resulting direction with the

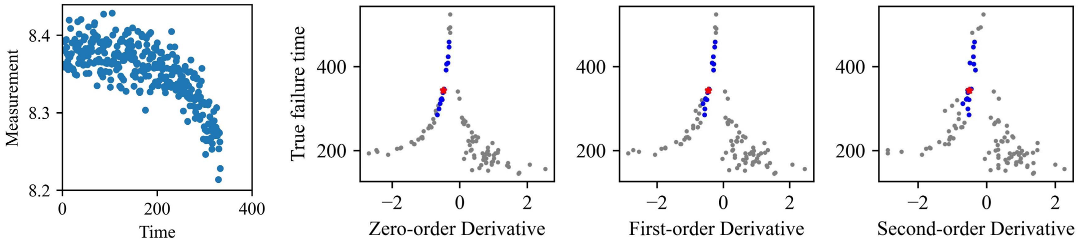


Fig. 6. Illustration of the extracted features from sensor BPR for in-service unit #94 of Dataset 3. The first plot shows the available sensor measurements. The next three plots show the true failure time versus three extracted features for the in-service unit (red points) and the historical units (blue points and grey points), where the blue points denote the 15 nearest neighbors.

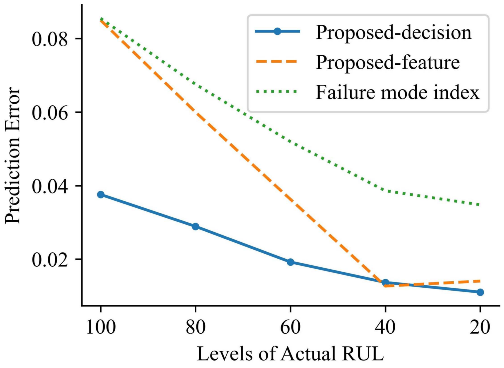


Fig. 7. Failure time prediction error for the in-service units of Dataset 3 using the failure mode index benchmark method and the proposed method with the decision-level fusion and feature-level fusion.

maximum variance best explains the failure time, and other directions are contaminated by noise. However, with high level of noise in $\boldsymbol{X}^{(p)}$, the variance of "other directions" can be large as well such that it interferes with the direction of interest that explains the failure time. In contrast, the decision-level fusion does not directly deal with the features, and by using the prediction variance $\hat{v}_j$, we can well control the weights for sensors with high uncertainty. It is worth mentioning that although feature-level fusion may underperform decision-level fusion in certain situations, it still reaches better performance than the existing benchmark method. It also has a great advantage in visualization. Specifically, in feature-level fusion, each unit can be characterized by a low-dimensional feature z_i, which can be visualized as in Fig. 4 and Fig. 6 to intuitively understand the data and assess the model adequacy. In addition, the final failure time is predicted directly by an LLR model in feature-level fusion, and thus the prediction variance can be quantified as in Section II.B. In contrast, for decision-level fusion, it is challenging to quantify the prediction variance without modeling signal correlations, according to (12).

D. Sensitivity to Sparse Data

In this subsection, we further explore the prognostic performance of the proposed method under cases of missing sensor measurements and sparse historical units. For demonstration purpose, we mainly focus on analyzing sensor T50 of Dataset 1 using degree 2 monotonic B-spline with two knots for feature extraction and the k-nearest-neighbor kernel function as in (6) with $n = 15$ for constructing the LLR.

At first, we consider missing sensor measurements. Specifically, for the T50 sensor signal of each historical and in-service unit, we randomly drop a fraction of measurements according to a signal missing rate r_m. The remaining measurements are then used to predict the failure time of the in-service units. Under each level of the signal missing rate r_m, this procedure is repeated for 50 times to obtain the average failure time prediction error. Note that by randomly dropping some sensor measurements, we are considering the case of unequally spaced and asynchronous signals across different units. The result is shown in Fig. 8, where different curves represent the set of in-service units with different levels of actual RUL. Generally, the prediction error increases with higher missing rate, which aligns with our intuition since less information is available. In addition, even with a missing rate as high as 0.7, the prognostic performance does not deteriorate too much comparing with the case of no missing data, which indicates our method to be relatively insensitive to missing sensor measurements, especially for in-service units with medium or relatively small actual RUL. One possible reason is that with monotonic B-spline of degree 2, the fitted sensor signals and extracted features are insensitive to sparse data. It also verifies the performance of our method in the case of unequally spaced measurements and asynchronous signals. Please note that we only consider the missing signal measurements to be random in this work. The prognostic performance of our method may be different under cases of non-random missing data with special patterns.

Next, we consider the case of sparse historical units. Specifically, we randomly drop a number of historical units, and only rely on the remaining historical units for failure time prediction of the in-service units. This procedure is repeated for 50 times. The average failure time prediction error for the in-service units is shown in Fig. 9, where the x-axis means the available number of historical units. It can be observed that even with only 40 historical units available, the prediction error is still quite stable, which shows the robustness of our method under sparse historical units. In this experiment, n is fixed to be 15. In practice, a smaller n may be used when less historical units are available. The selection of n will be further discussed in the next subsection.

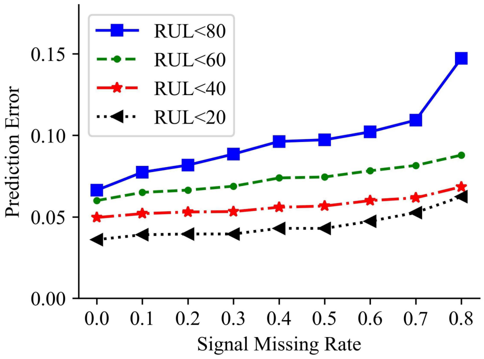


Fig. 8. Failure time prediction error for the in-service units of Dataset 1 using sensor T50 with different signal missing rates.

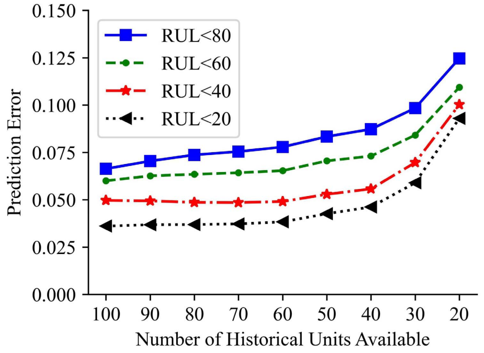


Fig. 9. Failure time prediction error for the in-service units of Dataset 1 using sensor T50 with different numbers of available historical units.

We also conducted simulation studies to investigate the sensitivity of feature-level fusion to the sparsity of historical units. The conclusion is similar as in Fig. 9 and thus is omitted.

E. Parameter Tuning

There are several parameters involved in our method. In particular, a kernel function $\mathcal{K}(\cdot)$ and the bandwidth n need to be specified for the single sensor module. The decision-level fusion involves two parameters q and γ. The feature-level fusion involves q and n', and after the fused feature z_i is obtained, a kernel function needs to be selected to construct the LLR.

Cross validation is commonly used for parameter tuning, which treats each historical unit as testing data and other historical units as training data in turn to evaluate the prediction error for all historical units. Parameters are usually selected to minimize the cross-validation prediction error. The limitation of cross validation is that it does not take the in-service unit into consideration. In fact, in this study, minimizing the prediction error at the in-service unit is more important than minimizing the overall cross-validation prediction error. Therefore, another possible strategy for parameter tuning is to minimize the failure time prediction variance v of the LLR model. However, a limitation of this strategy is that the estimated failure time prediction error $\hat{v}$ may be highly sensitive to the bandwidth n of the kernel function. For example, if the k-nearest-neighbor kernel function with $n = 4$ is used to construct the LLR, the fitted linear model will perfectly interpolate the four historical units, and thus the estimated prediction variance $\hat{v}$ will be 0 due to the lack of degree of freedom according to (9)-(11), which is obviously an under-estimation. Therefore, in this case study, we combine the two strategies for parameter tuning.

At first, we adopt the leave-one-out cross validation to select the kernel function and the bandwidth n for each in-service unit and each sensor. Specifically, given an in-service unit, we can extract features $\boldsymbol{x}_i$ from each historical unit i. Leave-one-out cross-validation is conducted based on $\{(\boldsymbol{x}_i, T_i), i \in \mathcal{M}\}$, where T_i is the actual failure time. Then, the failure time prediction error $e = \sum_{i\in\mathcal{M}}\left|\hat{T}_i - T_i\right|$ can be obtained, where $\hat{T}_i$ is the predicted failure time for unit i by treating $\boldsymbol{x}_i$ as the testing data and $\{(\boldsymbol{x}_k, T_k), k \in \mathcal{M}, k \neq i\}$ as the training data. The kernel function and the bandwidth n that lead to the smallest failure time prediction error e are selected. Since decision-level fusion does not provide an estimation for the variance of the final predicted failure time, we again rely on the leave-one-out cross validation for determining the parameters q and γ. For the feature-level fusion, given the kernel function with bandwidth n for constructing the LLR based on the fused feature z_i, we can select q and n' that lead to the minimum variance of the final predicted failure time for the in-service unit. Then, the leave-one-out cross-validation prediction error can be computed. The best set of parameters is selected to minimize the cross-validation prediction error.

V. Conclusion

Degradation modeling and prognostics is critical for predictive maintenance to avoid unexpected failure of engineering systems, which has attracted much research interest in recent years. However, most of the existing studies in the literature are restrictive and cannot fully address the following challenges in practice, which include (1) how to handle multiple sensors; (2) how to capture the complex and nonlinear relations between sensor signals and the degradation process; (3) how to consider multiple failure modes; (4) how to deal with unequally spaced sensor measurements and asynchronous signals; and (5) how to enhance the model interpretability for practitioners.

To address these challenges, in this paper, we propose a systematic, widely applicable method for degradation modeling and prognosis based on one or multiple sensors. The key idea is to predict the failure time of an in-service unit based on a subset of the nearest historical units. Specifically, for a single sensor, we extract features from the in-service unit and the historical units based on monotonic B-splines. Since the features are extracted based on the signal path instead of sensor measurements, our method can deal with asynchronous and unequally spaced sensors signals. Then, local linear models are built to characterize the nonlinear relation between extracted features and the failure time without any assumption on the specific functional form. Consequently, our method is flexible to deal with multiple failure modes. In addition, we propose to use the prediction variance as the goodness-of-fit measure for the local linear model. Then for multiple sensors, we combine the sensors by decision-level fusion or feature-level fusion, where the prediction variance is used to weight different sensors. Our proposed method is interpretable because the extracted features have clear meanings, and the local linear model can be easily visualized and explained. A case study with two datasets is conducted, and the proposed method achieves good performance with little preprocessing of the data, especially when the in-service units are close to failure.

In the future, there are several topics that worth further investigation. First, it is desired to explore the theoretical analysis on the relationship between the proposed method and some parametric models such as the general path model [7]. In parametric models, the relation between failure time and the features of the signal (not necessary the features mentioned in this paper) is assumed to have specific forms, while in the proposed method, no such assumption is made but we use local linear models to approximate the relation. Despite the difference, there are connections between the proposed method and some parametric models. Understanding the connections will provide more insights and stimulate follow-up studies. Second, due to the noise and lack of data, the performance of the proposed method may degrade, especially when the in-service unit is in the beginning of the degradation process. In this case, parametric models may achieve better performance. Therefore, a semi-parametric method can be developed to combine the proposed method with parametric models to alleviate this issue. Last, we only consider units under one operational condition in this study. In the future, how to consider multiple operational conditions is worth further investigation.

Appendix
Monotonic B-Spline With Infinite Support

Let $\xi_0 < \xi_1 < \cdots < \xi_{K-1}$ be the knots, and d be the degree of the B-spline. In addition, let $\{\psi_{k,d}(t), k = 0, \cdots, K+d\}$ be the basic functions of degree d. Comparing with the notation of Section II.A, here for simplicity, we omit the subscript i which denotes the unit, and add the subscript d to denote the degree. The B-spline basis functions of degree 0 are defined as

$$\begin{aligned}\psi_{0,0}(t) &= I_{(-\infty,\xi_0)}(t),\\ \psi_{K,0}(t) &= I_{[\xi_{K-1},+\infty)}(t),\\ \psi_{k,0}(t) &= I_{[\xi_{k-1},\xi_k)}(t), \quad 1 \le k \le K-1,\end{aligned}$$

where $I_{(\cdot)}$ denotes the indicator function. The induction formulas for the B-spline of degree $1 \le d \le K$ are:

$$\begin{aligned}\psi_{0,d}(t) &= \frac{\xi_0 - t}{C_0}\psi_{0,d-1}(t),\\ \psi_{k,d}(t) &= \psi_{k-1,d-1}(t) + \frac{\xi_k - t}{C_0}\psi_{k,d-1}(t), \quad 1 \le k < d\\ \psi_{d,d}(t) &= \begin{cases}\psi_{d-1,d-1}(t) + \dfrac{\xi_d - t}{\xi_d - \xi_0}\psi_{d,d-1}(t), & \text{if } K > d\\ \psi_{d-1,d-1}(t) + \psi_{d,d-1}(t), & \text{if } K = d\end{cases}\\ \psi_{k,d}(t) &= \frac{t - \xi_{k-d-1}}{\xi_{k-1} - \xi_{k-d-1}}\psi_{k-1,d-1}(t) + \frac{\xi_k - t}{\xi_k - \xi_{k-d}}\psi_{k,d-1}(t),\\ &\qquad d+1 \le k < K\\ \psi_{K,d}(t) &= \psi_{K,d-1}(t) + \frac{t - \xi_{K-d-1}}{\xi_{K-1} - \xi_{K-d-1}}\psi_{K-1,d-1}(t),\\ &\qquad \text{if } K > d\\ \psi_{k,d}(t) &= \frac{t - \xi_{k-d-1}}{C_1}\psi_{k-1,d-1}(t) + \psi_{k,d-1}(t),\\ &\qquad K+1 \le k < K+d\\ \psi_{K+d,d}(t) &= \frac{t - \xi_{K-1}}{C_1}\psi_{K+d-1,d-1}(t).\end{aligned}$$

In this study, we use $C_0 = C_1 = \frac{\xi_{K-1} - \xi_0}{K-1}$. In addition, let $\psi_{k,d}^{(p)}(t)$ be the corresponding pth-order derivative ($p \ge 1$), i.e., $\psi_{k,d}^{(1)}(t) = \psi'_{k,d}$, $\psi_{k,d}^{(2)}(t) = \psi''_{k,d}$. The derivatives can be calculated as

$$\begin{aligned}\psi_{k,d}^{(p)}(t) &= -\frac{d}{C_0}\psi_{k,d-1}^{(p-1)}(t), \quad 0 \le k < d\\ \psi_{d,d}^{(p)}(t) &= -\frac{d}{\xi_d - \xi_0}\psi_{d,d-1}^{(p-1)}(t),\\ \psi_{k,d}^{(p)}(t) &= \frac{d}{\xi_{k-1} - \xi_{k-d-1}}\psi_{k-1,d-1}^{(p-1)}(t) - \frac{d}{\xi_k - \xi_{k-d}}\psi_{k,d-1}^{(p-1)}(t),\\ &\qquad d+1 \le k < K\\ \psi_{K,d}^{(p)}(t) &= \frac{d}{\xi_{K-1} - \xi_{K-d-1}}\psi_{K-1,d-1}^{(p-1)}(t),\\ \psi_{k,d}^{(p)}(t) &= \frac{d}{C_1}\psi_{k-1,d-1}^{(p-1)}(t), \quad K+1 \le k \le K+d\end{aligned}$$

More details can be found in [30], [36].

References

[1] H. Ye, X. Wang, and K. Liu, "Adaptive preventive maintenance for flow shop scheduling with resumable processing," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 1, pp. 106–113, Jan. 2021.

[2] J. Guo, Z. Li, and M. Li, "A review on prognostics methods for engineering systems," *IEEE Trans. Rel.*, vol. 69, no. 3, pp. 1110–1129, Sep. 2020.

[3] Y. Hong, M. Zhang, and W. Q. Meeker, "Big data and reliability applications: The complexity dimension," *J. Qual. Technol.*, vol. 50, no. 2, pp. 135–149, Apr. 2018.

[4] Z.-S. Ye and M. Xie, "Stochastic modelling and analysis of degradation for highly reliable products," *Appl. Stochastic Models Bus. Ind.*, vol. 31, no. 1, pp. 16–32, 2015.

[5] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation—A review on the statistical data driven approaches," *Eur. J. Oper. Res.*, vol. 213, no. 1, pp. 1–14, 2011.

[6] K. Liu and J. Shi, "Internet of Things (IoT)-enabled system informatics for service decision making: Achievements, trends, challenges, and opportunities," *IEEE Intell. Syst.*, vol. 30, no. 6, pp. 18–21, Nov. 2015.

[7] C. J. Lu and W. O. Meeker, "Using degradation measures to estimate a time-to-failure distribution," *Technometrics*, vol. 35, no. 2, pp. 161–174, May 1993.

[8] N. Gebraeel, "Sensory-updated residual life distributions for components with exponential degradation patterns," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 382–393, Oct. 2006.

[9] L. Hao, K. Liu, N. Gebraeel, and J. Shi, "Controlling the residual life distribution of parallel unit systems through workload adjustment," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1042–1052, Apr. 2017.

[10] Z.-S. Ye, Y. Wang, K.-L. Tsui, and M. Pecht, "Degradation data analysis using Wiener processes with measurement errors," *IEEE Trans. Rel.*, vol. 62, no. 4, pp. 772–780, Dec. 2013.

[11] X. Wang and D. Xu, "An inverse Gaussian process model for degradation data," *Technometrics*, vol. 52, no. 2, pp. 188–197, May 2010.

[12] R. Kontar, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, "Nonparametric modeling and prognosis of condition monitoring signals using multivariate Gaussian convolution processes," *Technometrics*, vol. 60, no. 4, pp. 484–496, Oct. 2018.

[13] A. Chehade and K. Liu, "Structural degradation modeling framework for sparse data sets with an application on Alzheimer's disease," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 192–205, Jan. 2019.

[14] Y. Lin, K. Liu, E. Byon, X. Qian, S. Liu, and S. Huang, "A collaborative learning framework for estimating many individualized regression models in a heterogeneous population," *IEEE Trans. Rel.*, vol. 67, no. 1, pp. 328–341, Mar. 2017.

[15] C. Song, K. Liu, and X. Zhang, "Integration of data-level fusion model and kernel methods for degradation modeling and prognostic analysis," *IEEE Trans. Rel.*, vol. 67, no. 2, pp. 640–650, Jun. 2018.

[16] M. Kim, C. Song, and K. Liu, "A generic health index approach for multisensor degradation modeling and sensor selection," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1426–1437, Jul. 2019.

[17] C. Song and K. Liu, "Statistical degradation modeling and prognostics of multiple sensor signals via data fusion: A composite health index approach," *IISE Trans.*, vol. 50, no. 10, pp. 853–867, Oct. 2018.

[18] X. Fang, N. Z. Gebraeel, and K. Paynabar, "Scalable prognostic models for large-scale condition monitoring applications," *IISE Trans.*, vol. 49, no. 7, pp. 698–710, 2017.

[19] K. Medjaher, D. A. Tobon-Mejia, and N. Zerhouni, "Remaining useful life estimation of critical components with application to bearings," *IEEE Trans. Rel.*, vol. 61, no. 2, pp. 292–302, Jun. 2012.

[20] C. Su and J. Shen, "A novel multi-hidden semi-Markov model for degradation state identification and remaining useful life estimation," *Qual. Rel. Eng. Int.*, vol. 29, no. 8, pp. 1181–1192, Dec. 2013.

[21] C. Song, K. Liu, and X. Zhang, "A generic framework for multisensor degradation modeling based on supervised classification and failure surface," *IISE Trans.*, vol. 51, no. 11, pp. 1288–1302, Nov. 2019.

[22] X. Fang, H. Yan, N. Gebraeel, and K. Paynabar, "Multi-sensor prognostics modeling for applications with highly incomplete signals," *IISE Trans.*, vol. 53, no. 5, pp. 597–613, Aug. 2020.

[23] A. Chehade, C. Song, K. Liu, A. Saxena, and X. Zhang, "A data-level fusion approach for degradation modeling and prognostic analysis under multiple failure modes," *J. Qual. Technol.*, vol. 50, no. 2, pp. 150–165, Apr. 2018.

[24] M. Kim and K. Liu, "A Bayesian deep learning framework for interval estimation of remaining useful life in complex systems by incorporating general degradation characteristics," *IISE Trans.*, vol. 53, no. 3, pp. 326–340, Mar. 2021.

[25] Z. Tian, "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring," *J. Intell. Manuf.*, vol. 23, no. 2, pp. 227–237, Apr. 2012.

[26] T. H. Loutas, D. Roulias, and G. Georgoulas, "Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic ϵ-support vectors regression," *IEEE Trans. Rel.*, vol. 62, no. 4, pp. 821–832, Dec. 2013.

[27] L. L. Li, D. J. Ma, and Z. G. Li, "Residual useful life estimation by a data-driven similarity-based approach," *Qual. Rel. Eng. Int.*, vol. 33, no. 2, pp. 231–239, Mar. 2017.

[28] Q. Zhang, P. W.-T. Tse, X. Wan, and G. Xu, "Remaining useful life estimation for mechanical systems based on similarity of phase space trajectory," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2353–2360, 2015.

[29] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.

[30] S. Jahani, R. Kontar, S. Zhou, and D. Veeramani, "Remaining useful life prediction based on degradation signals using monotonic B-splines with infinite support," *IISE Trans.*, vol. 52, no. 5, pp. 537–554, May 2020.

[31] J. O. Ramsay, "Monotone regression splines in action," *Stat. Sci.*, vol. 3, no. 4, pp. 425–461, Nov. 1988.

[32] M. C. Meyer, "Inference using shape-restricted regression splines," *Ann. Appl. Statist.*, vol. 2, no. 3, pp. 1013–1033, Sep. 2008.

[33] C. Loader, *Local Regression and Likelihood*. New York, NY, USA: Springer, 1999.

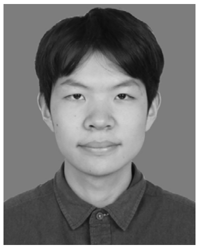
[34] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proc. IEEE*, vol. 85, no. 1, pp. 6–23, Jan. 1997.

[35] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, Denver, CO, USA, Oct. 2008, pp. 1–9.

[36] S. Corlay, "B-spline techniques for volatility modeling," *J. Comput. Finance*, vol. 19, no. 3, pp. 97–135, Mar. 2016.

Changyue Song (Member, IEEE) received the B.S. degree in industrial engineering from Tsinghua University, Beijing, China, in 2012, the M.S. degree in management science and engineering from Peking University, Beijing, in 2015, and the M.S. degree in statistics and the Ph.D. degree in industrial engineering from the University of Wisconsin–Madison, Madison, WI, USA, in 2018 and 2020, respectively.

Currently, he is an Assistant Professor with the School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ, USA. His research interests include data analytics for system improvement, process modeling and prognosis, and statistical learning.



Ziqian Zheng received the B.E. degree in automation engineering from Xi'an Jiaotong University, Xi'an, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Industrial and Systems Engineering, University of Wisconsin–Madison, Madison, WI, USA.

His research interests include data analytics for system prognosis and big data monitoring.

Kaibo Liu (Senior Member, IEEE) received the B.S. degree in industrial engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2009, and the M.S. degree in statistics and the Ph.D. degree in industrial engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2011 and 2013, respectively.

He is currently an Associate Professor with the Department of Industrial and Systems Engineering, University of Wisconsin–Madison, Madison, WI, USA. His research interests include data fusion for process modeling, monitoring, diagnosis, and prognostics, and decision making.

Dr. Liu is a member of ASQ, INFORMS, IISE, and SME.