1-19-2022

A Neural Network based Proportional Hazard Model for IoT Signal Fusion and Failure Prediction

Yuxin Wen
Chapman University, yuwen@chapman.edu

Xingxin Guo
Peking University

Junbo Son
University of Delaware

Jianguo Wu
Peking University

Recommended Citation

Yuxin Wen, Xingxin Guo, Junbo Son & Jianguo Wu (2022) A Neural Network based Proportional Hazard Model for IoT Signal Fusion and Failure Prediction, IISE Transactions, https://doi.org/10.1080/24725854.2022.2030881

A Neural Network based Proportional Hazard Model for IoT Signal Fusion and Failure Prediction

A Neural Network based Proportional Hazard Model for IoT Signal Fusion and Failure Prediction

Yuxin Wen[a], Xingxin Guo[b], Junbo Son[c], Jianguo Wu[b*]

[a] Fowler School of Engineering, Chapman University, Orange, CA, USA
[b] Department of Industrial Engineering and Management, College of Engineering, Peking University, Beijing, China
[c] Department of Business Administration, Alfred Lerner College of Business & Economics, University of Delaware, Newark, DE, USA
[*] Corresponding author, e-mail: j.wu@pku.edu.cn

Abstract

Accurate prediction of remaining useful life (RUL) plays a critical role in optimizing condition-based maintenance decisions. In this paper, a novel joint prognostic modeling framework that simultaneously combines both time-to-event data and multi-sensor degradation signals is proposed. With the increasing use of IoT devices, unprecedented amounts of diverse signals associated with the underlying health condition of *in-situ* units have become easily accessible. To take full advantage of the modern IoT-enabled engineering systems, we propose a specialized framework for RUL prediction at the level of individual units. Specifically, a Bayesian linear regression model is developed for the multi-sensor degradation signals and a functional neural network is proposed to allow the proportional hazard model to characterize the complex non-linearity between the hazard function and degradation signals. Based on the proposed model, an online model updating procedure is established to accurately predict RUL in real time. The advantageous features of the proposed method are demonstrated through simulation studies and the application to a high-fidelity gas turbine engine dataset.

***Keywords*:** Cox PH model; degradation data; joint prognostic model; neural networks; remaining useful life prediction.

1 Introduction

Condition-based maintenance system monitors various signals that are closely related to the underlying health condition of *in-situ* units to determine how and when to take a maintenance intervention. It has shown great potentials for preventing unexpected failures and reducing operational costs in diverse engineering applications (Pecht 2010). Recent advances in the Internet of Things (IoT) have further realized cost effective and accurate real-time online monitoring, which provides a big window of opportunities to researchers. The sensors embedded in IoT devices continuously collect data and transmit them through a wireless communication network to data processing center for rigorous analysis. The unprecedented data

availability enables sophisticated data fusion and allows us to develop accurate and robust algorithms to predict the remaining useful life (RUL), or the time of imminent failure, so that the optimal maintenance decisions can be made.

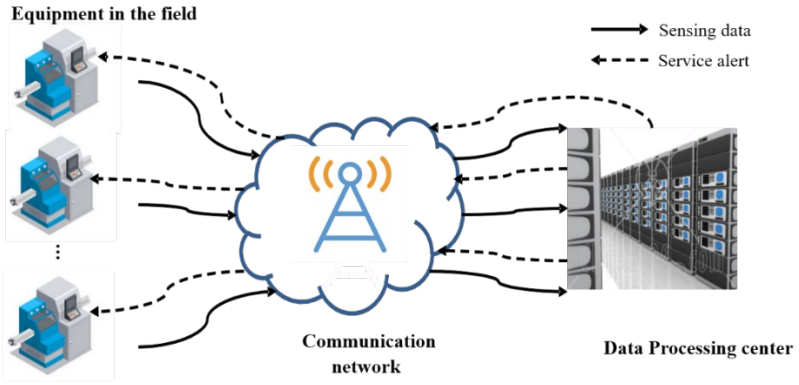

Figure 1. Structure of IoT-enabled systems

The RUL is defined as the time left for a unit before it completely stops functioning or before it fails to meet the operational requirement (Jardine, Lin, and Banjevic 2006). In the past few decades, various degradation modeling and RUL prediction methods have been developed. These prognostic models, in general, can be categorized into physics-based and data-driven approaches (Jardine, Lin, and Banjevic 2006; Heng *et al.* 2009). The physics-based approaches assume well-defined physical models for the underlying degradation process (Pecht 2010). Despite some promising features, however, it is often challenging to develop an adequate physical model because of the highly complex degradation mechanisms of modern engineering systems. Therefore, data-driven approaches have been increasingly appreciated in many applications. Many components in modern engineering systems are IoT devices equipped with sensory devices and wireless communication capabilities; hence, the degradation signals, a key for successful implementation of data-driven methods, are abundantly available (Si *et al.* 2011). Figure 1 shows the structure of an IoT-enabled system that can be found in various applications.

There are two types of failures considered in the existing data-driven prognostic models: soft failure and hard failure (Son *et al.* 2013). Soft failure is defined as an event when a degradation signal hits a pre-specified failure threshold for the first time (Yu and Fuh 2010). Once a soft failure occurs, the performance

of the unit no longer meets the working requirement, though the unit may still be working. In such a case, a mathematical model, e.g., mixed-effects model or stochastic process, is commonly applied to characterize the evolution of degradation signals and to predict when the degradation signals hit the failure threshold (Zio and Di Maio 2010). To accurately model the propagation paths of various degradation signals, researchers have used linear or nonlinear regression models often with random parameters to capture the unit-to-unit variability (Gebraeel *et al.* 2005). Instead of regression models, stochastic processes, such as Gamma process (Ye *et al.* 2014), Wiener process (Wen *et al.* 2018a), and inverse Gaussian process (Peng *et al.* 2014), have also been used to account for the temporal variation of degradation signals. In many cases, these models assume that the failure threshold on degradation signals is constant and deterministic for all units. However, a fixed failure threshold may not be sufficient to characterize the health status of all individual units due to their heterogeneous features. Indeed, some studies in the literature tried to address the unit-to-unit heterogeneity issue (Wang and Coit 2007; Liang *et al.* 2011; Chehade, Bonk, and Liu 2017). For instance, the concept of random thresholds have been proposed to allow each unit to have its own failure threshold (Wang and Coit 2007; Yu and Fuh 2010).

The other type of failure, namely the hard failure, is closely related to the degradation signal but not solely defined by it (Liu *et al.* 2013). Rather, under the definition of hard failure, the worsening condition of a system increases the hazard (or risk of failure) and the actual system failure occurs probabilistically based on the risk level. Therefore, when hard failure happens, a unit completely stops working. To capture this risk-based failure mechanism, the time-to-event (time-to-failure) data is used to estimate the probability distribution of survival times, e.g., Weibull distribution or Cox Proportional Hazards (PH) models (Cox 1972; Lawless 2011; Meeker and Escobar 2014). The hard failure prediction methods do not need the failure threshold assumption on degradation signals. Moreover, various time-invariant information about individual units (such as built-year, manufacturer, and product type) can be easily incorporated due to the flexible model structure. However, modeling the probability distribution of failure times has been done primarily at the population level rather than at the individual unit level (McPherson 2018).

From our discussion above, it is easy to foresee the benefit of combining both time-to-event data and degradation signals. As a matter of fact, in the fields of clinical science and biostatistics, jointly modeling both degradation signals and time-to-event data has been well studied (Tsiatis, Degruttola, and Wulfsohn 1995). The degradation signals (often referred to biomarkers in medical domain) are observations collected from patients that are closely related to the patients' underlying health condition. Under the joint modeling framework, typically, the degradation signals are modeled by a mixed-effects model and time-to-event data are modeled by the Cox PH model. The Cox PH model is one of the most popular methods for estimating the probability distribution of survival time based on one or more predictor variables. Well-known examples of joint modeling include evaluating new treatments in AIDS through CD4-lymphocyte counts (Tsiatis, Degruttola, and Wulfsohn 1995) and prostate cancer detection/treatment by monitoring the prostate specific antigen level (Yu, Taylor, and Sandler 2008). A comprehensive review about the joint modeling with clinical applications can be found in Tsiatis and Davidian (2004). On the other hand, the literature on utilizing the joint modeling approach to failure prediction in engineering field is scarce. Liao, Zhao, and Guo (2006) presented a PH model and logistic regression model to predict the RUL of bearings. Zhou *et al.* (2014) proposed a joint modeling framework to predict the RUL of automotive lead-acid batteries. Later, this work was extended by introducing Wiener process into the joint modeling framework (Man and Zhou 2018). Yan *et al.* (2018) proposed a functional principal components analysis based Cox PH model. By using a moving time window, their Cox PH model can take advantage of historical observations to address the well-known biased estimation issue existed in many existing Cox PH models where only the latest observation is used (Kalbfleisch and Prentice 2011). Recently, Yue and Kontar (2020) developed a joint multivariate Gaussian convolution process coupled with Cox PH model. Despite the noticeable advancements, however, the existing joint modeling approaches linearly combine the multiple covariates, which may not be suitable to capture highly nonlinear relationship among covariates.

Unlike the engineering field, in clinical applications, several extensions of Cox model using neural network (NN) have been proposed to improve the prognostic accuracy. Various types of NN have been

highlighted in the literature due to their flexible data fusion ability and capabilities on approximating arbitrary functions. Faraggi and Simon (1995) is the first to propose a NN-based Cox model. In their work, a single logistic hidden layer feed-forward NN was developed to fuse multiple covariates in the hazard function. Later, several variants of the NN-based Cox model have appeared in the clinical literature taking advantage of new NN architectures, larger data sets, and better optimization strategies, e.g., Cox-nnet (Ching, Zhu, and Garmire 2018), SurvivalNet (Yousefi *et al.* 2017), and DeepSurv (Katzman *et al.* 2018). These models focus on time-independent or static covariates, e.g., clinical and genetic features, trying to explore the effectiveness of various treatment options. In this field, one of the recent advancements was achieved by Kvamme, Borgan, and Scheel (2019). In their paper, the Cox PH model was integrated with NNs and allowed the NNs to model interactions between the covariates and time-to-event data by treating the longitudinal data as a regular covariate. Although this approach successfully makes the model more flexible by relaxing the proportionality assumption of classical Cox PH model, still, it cannot handle time-dependent covariates (i.e., longitudinal degradation signals in the context of condition-based maintenance).

To some extent, these NN-based Cox models can be used for RUL prediction by treating the observations (degradation signals) as time-dependent covariates. However, directly using longitudinal observations as time-dependent covariates in Cox model is known to have two major sources of bias (Tsiatis and Davidian 2001). First, the parameter estimates can be biased due to the "last-observation-carried-forward" (LOCF) inference method (Arisido et al. 2019). As the name suggests, LOCF is a naïve approach that is frequently used for Cox model with time-dependent covariates. This method substitutes the latest observation for the true time-dependent covariates instead of estimating it with the set of entire historical measurements. Such an over-simplified strategy inevitably leads to biased estimation. The issue becomes more critical when data collection occurs irregularly during the monitoring period (Therneau, Crowson, and Atkinson 2013). In practice, the data collection plan may change over time depending on the health condition. Second, it is challenging to account for measurement errors in the observations. When measurement error is present, failure to address it may easily introduce bias into the inference process (Prentice 1982; Dupuy and Mesbah

2002). Thus, using longitudinal observations directly as time-dependent covariates in Cox model may not be a viable option for accurately predicting the survival curve. Because of this limitation, the existing methods cannot take the full advantage of IoT-enabled real-time online RUL prediction for individual in-service units.

To fill this research gap, we develop an NN-based joint prognostic modeling framework for the failure prediction at the individual unit level by jointly modeling the time-to-event data and degradation signals. The proposed approach models both the population characteristics and the individual heterogeneity of degradation signals while avoiding undesirable bias introduced to the estimation process. Specifically, we model the multi-sensor degradation signals by a linear mixed-effects model. Then, the predicted values of the degradation signals provided by the mixed-effects model are fed into NN-based Cox model. The overall process has two distinctive stages: offline parameter estimation stage and online model updating and prediction stage. In the offline stage, model parameters that are associated with population-level characteristics, e.g., baseline hazard and weights for the NN, and parameters that represent individual-level variability for degradation signals are estimated. In the online stage, population-level parameters are treated as fixed and individual-level parameters will be updated over time as more data from the specific unit/sensor of interest are collected. At the last step, the RUL prediction can be made for the in-service unit. To the best of our knowledge, our work is the first to incorporate multiple degradation signals and time-to-event data simultaneously using NNs and Cox PH model especially for the individualized RUL prediction. The major contribution of this paper lies in threefold: (1) a new joint prognostic modeling framework is proposed which combines both multiple degradation signals and time-to-event data to predict the RUL of an individual unit; (2) a novel NN is developed for improving the Cox PH model's capability of modeling nonlinear relationship among multiple covariates; (3) a two-stage process with offline modeling and online prediction stages specifically designed for our NN-based Cox model is established. By using the Bayesian approach, our method can sequentially update the RUL distribution as new signals from an IoT-equipped in-service unit become available; hence, making the prediction result highly individualized.

The rest of this paper is organized as follows. A detailed description of the proposed method is provided in Section 2, including each step of offline degradation signal modeling, NN training, online Bayesian updating, and real-time RUL prediction. Section 3 demonstrates the effectiveness of our proposed method through a series of simulations and reports the performance evaluation results within the application to gas turbine engine maintenance. Finally, Section 4 concludes the paper with a discussion of potential future work.

2 Advanced Joint Prognostic Model Integrated with Neural Network

The overall idea of the proposed method is to improve the joint prognostic modeling framework by introducing the NN into the model structure. It provides greater flexibility in boosting the prediction accuracy by adequately modeling nonlinear relationships among multiple degradation signals. Our framework is composed of two stages: (1) the offline stage for multi-sensor degradation signal modeling and NN-based Cox model training, and (2) the online stage for sequential degradation model updating and RUL prediction. At the offline stage, the model involves two sub-models: a mixed-effects model for the multi-sensor degradation modeling, an NN-based Cox model for time-to-event modeling and degradation data fusion. Specifically, the mixed-effects model is employed to capture the evolution of degradation signals and to predict future degradation signal propagation paths. The predicted degradation signals obtained from the mixed-effects model are then included into the NN-based Cox model as time-dependent covariates for the hazard rate estimation. The online stage starts as we monitor a specific in-service unit and collect multi-sensor degradation signals from that unit. A Bayesian updating scheme is used to compute the posterior distributions of the mixed-effects model parameters for the in-service unit, and then a real-time RUL prediction of the unit is made based on the updated parameters and the NN model accordingly. One of the main advantages of the Bayesian framework is its capability of quantifying the uncertainties stemming from the stochastic nature of degradation processes in the form of posterior distribution. The initial (prior) distribution for the model parameters reflects the overall population-level characteristics. As more data become available, posterior distribution can be obtained for the parameters; hence, the model

can be individualized tailored to the specific unit of our interest. While doing so, Bayesian updating method naturally improves the prediction accuracy. For a specific in-service unit, in its early life cycle when observations are scarce, the Bayesian framework borrows information from the population by using the prior distribution (possibly with large variability), then gradually updates the posterior distributions as more observations accumulate. The technical details are provided in the following subsections and a step-by-step illustration of the proposed joint modeling framework is shown in Figure 2.

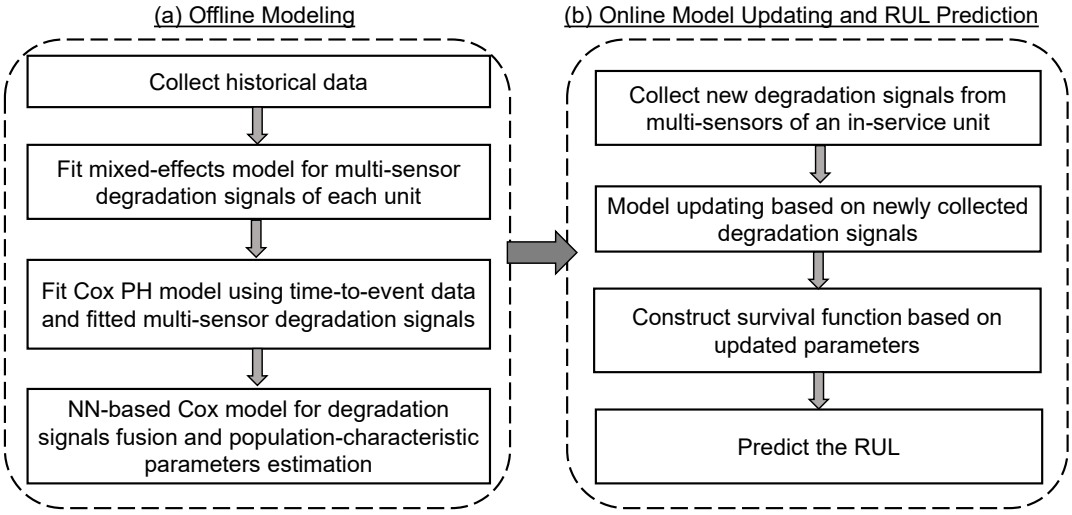

Figure 2. Illustration of the proposed framework

2.1 Joint Modeling Framework for Degradation Signals and Time-to-event Data

Suppose N historical units have been collected, and each unit i has J degradation signals with n_i observations given by

$$\boldsymbol{Y}_i = \begin{bmatrix} y_{i,1,1} & y_{i,2,1} & \cdots & y_{i,J,1} \\ y_{i,1,2} & y_{i,2,2} & \cdots & y_{i,J,2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{i,1,n_i} & y_{i,2,n_i} & \cdots & y_{i,J,n_i} \end{bmatrix}, \tag{1}$$

for $i = 1, \ldots, N$. We assume that all the J degradation signals of each unit are observed at the same time points for the sake of notational convenience. For unit i, its event time $F_i = min\{T_i, C_i\}$ is also recorded, where T_i denotes the failure time and C_i denotes the censoring time. Let δ_i denote an indicator function taking value 1 if the unit i is failed at T_i, or taking value 0 if it is censored at C_i. Now the full observed dataset for the ith unit can be denoted as $\boldsymbol{D}_i = \{\boldsymbol{Y}_i, F_i, \delta_i\}$. The proposed joint modeling approach involves

two sub-models: a mix-effects model for the degradation signals and an NN-based Cox model for the time-to-event data.

For the degradation signals collected from multiple sensors, linear regression models are typically employed to capture both the population trend and variability among heterogeneous units (Lu and Meeker 1993; Gebraeel *et al.* 2005; Chen and Tsui 2013; Wen, Wu, and Yuan 2017; Wen *et al.* 2018b). The degradation path from sensor j of unit i can be written as

$$y_{i,j}(t) = r_{i,j}(t) + \varepsilon_j \,, \tag{2}$$

where ε_j is assumed to be independent and identically distributed (*i.i.d.*) Gaussian noise with zero mean and variance σ_j^2. $r_{i,j}(t)$ is the true but unobservable value of the degradation signal j for unit i at time t, and it can be described as

$$r_{i,j}(t) = \boldsymbol{X}_{i,j,t}\boldsymbol{B}_{i,j} \,, \tag{3}$$

where $\boldsymbol{X}_{i,j,t}$ is a $q_j + 1$ dimensional vector defined as $\boldsymbol{X}_{i,j,t} = [1, t, t^2, \dots, t^{q_j}]$ and $\boldsymbol{B}_{i,j}$ is a $q_j + 1$ dimensional vector of random coefficients. Note that the order of polynomial regression could vary across different sensors and it can be determined by some quantitative model selection methods. In this paper, we use the Bayesian information criterion (BIC) to decide the best order of polynomial form for our model.

For the random coefficients $\boldsymbol{B}_{i,j}$, a multivariate normal distribution is commonly assumed, i.e., $\boldsymbol{B}_{i,j} \sim N\left(\boldsymbol{\mu}_0^{(j)}, \boldsymbol{\Sigma}_0^{(j)}\right)$, where $\boldsymbol{\mu}_0^{(j)}$ and $\boldsymbol{\Sigma}_0^{(j)}$ are the mean vector and the covariance matrix, respectively. Please note that the linear regression model can be replaced by any other models if the data show strong non-linear behaviors.

For the time-to-event data, we use the Cox PH model (Cox 1972), which is a widely used model in survival analysis. The Cox PH model is defined as:

$$h_i(t) = h_0(t)\, exp\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i\right), \tag{4}$$

where $h_0(t)$ is a baseline hazard function in either a non-parametric or parametric form, $\boldsymbol{\omega}$ is a coefficient vector, and $\boldsymbol{x}_i$ is a vector of covariates for unit *i*. The Cox PH model assumes that the hazard rate of a unit is proportional to its baseline hazard rate $h_0(t)$. To incorporate degradation signals into the Cox model,

$\boldsymbol{R}_i(t) = \left[r_{i,1}(t), r_{i,2}(t), \ldots, r_{i,J}(t)\right]^T$ is treated as time-dependent covariates and replace vector $\boldsymbol{x}_i$. By doing so, multiple degradation signals are combined linearly to the Cox model while accounting for the unit-to-unit heterogeneity. To include time-dependent covariates into the Cox model, using the raw signal observations as time-dependent covariates would probably be the most straightforward way. In fact, many studies in the literature have investigated this matter in various domains. Among them, several studies have found that such an approach suffers from noticeable bias due to improper inference (Tsiatis and Davidian 2001). Especially, the noise (measurement error) in the degradation signals is known as the major source of bias (Prentice 1982; Dupuy and Mesbah 2002). In our method, by using $r_{i,j}(t)$ instead of the raw signals $y_{ij}(t)$, we avoid undesirable bias introduced by external noise that may lead to poor prediction accuracy (Zhou *et al.* 2014). With the joint modeling approach, a mixed-effects model for the underlying degradation trajectory is linked to the survival model using shared random effects. This approach allows us to study the relationship between hazards and degradation processes without the last-observation-carried-forward assumption. One may suggest that the $\boldsymbol{B}_{i,j}$ can be directly used since the difference across units stems from the coefficient $\boldsymbol{B}_{i,j}$. However, the degradation signal propagates over time and the health condition should be properly characterized by both $\boldsymbol{B}_{i,j}$ and time (i.e., $\boldsymbol{X}_{i,j,t}$). In other words, only with both $\boldsymbol{B}_{i,j}$ and $\boldsymbol{X}_{i,j,t}$, can we dynamically capture the degradation status. As the underlying health status is closely related to degradation signals, $r_{i,j}(t)$ actually can be treated as a feature extracted from the coefficient $\boldsymbol{B}_{i,j}$ and time $\boldsymbol{X}_{i,j,t}$ based on the degradation mechanism. Although $\boldsymbol{B}_{i,j}$ and $\boldsymbol{X}_{i,j,t}$ collectively define the signal path, using $r_{i,j}(t)$ is a more straightforward way than directly feeding both $\boldsymbol{B}_{i,j}$ and $\boldsymbol{X}_{i,j,t}$ to the network.

With the hazard function in (4), the probability density function of the failure time of unit i can be calculated as

$$f_i(t) = h_i(t)^{\delta_i} S_i(t) = h_i(t)^{\delta_i} \, exp\left(\int_0^t -h_i(u)du\right), \tag{5}$$

where $S_i(t) = exp\left(\int_0^t -h_i(u)du\right)$ is the survival function. As we can see, the degradation signals are linked to the failure time distribution through a hazard function.

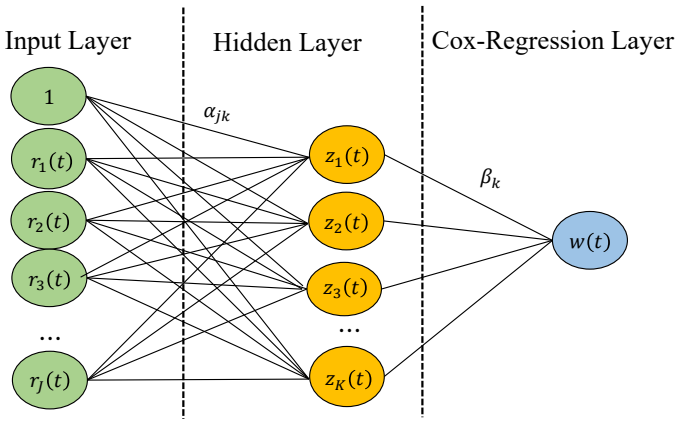

Figure 3. Illustration of the proposed NN-based Cox model

The Cox model assumes a linear combination of multiple degradation signals and this feature limits the suitability of modeling nonlinear interactions among prognostic factors. To overcome this well-known issue of Cox model, in this paper, a NN is used to fuse multi-sensor degradation signals in a non-linear manner. Figure 3 shows an example of a one-hidden-layer NN-based Cox model. NNs have shown great potential when complex interactions or non-linear effects of inputs exist. The NN architecture is composed of inputs, one or more fully connected hidden layers, and a Cox-Regression output layer. Note that the output layer is a linear combination of the last hidden layer without any activation function and bias (intercept). The bias term only scales the baseline hazard and would not contribute to the relative risk function (Kvamme, Borgan, and Scheel 2019).

The functional representation of the output of NN is

$$w_i(t) = \sum_{k=1}^{K} \beta_k z_{i,k}(t) = \sum_{k=1}^{K} \beta_k \left(A\left(\alpha_{0k} + \sum_{j=1}^{J} \alpha_{jk} r_{i,j}(t) \right) \right), \tag{6}$$

where $A(\cdot)$ represents an activation function, and α_{0k} is a bias weight that is used as an adjustable offset for the activation function. $\{\alpha_{jk}, \beta_k, j = 0,1,2, \dots, J, k = 1,2,3 \dots, K\}$ represent the NN weights. Here J is the number of input nodes which is equal to the number of sensors, K is the number of hidden nodes which is determined through training. Replacing the linear function $\boldsymbol{\omega}^T \boldsymbol{R}_i(t)$ of Cox model by the output of the NN, $w_i(t)$, the hazard function of unit i becomes

$$h_i(t|\boldsymbol{R}_i(t)) = h_0(t)\, exp\big(w_i(t)\big). \tag{7}$$

2.2 Offline Parameters Estimation

Denote all the model parameters as $\boldsymbol{\psi} = \left\{\boldsymbol{\mu}_0^{(j)}, \boldsymbol{\Sigma}_0^{(j)}, \sigma_j^2, h_0(t), \alpha_{jk}, \beta_k\right\}$ for $j = 0, 1, 2, \ldots, J$ and $k = 1, 2, \ldots, K$. Among them, $\{h_0(t), \alpha_{jk}, \beta_k\}$ represent the population characteristics, i.e., they are kept the same for all units from the same population. On the other hand, $\left\{\boldsymbol{\mu}_0^{(j)}, \boldsymbol{\Sigma}_0^{(j)}, \sigma_j^2\right\}$ capture the unit-to-unit variability for each sensor. In the offline modeling, all the parameters $\boldsymbol{\psi}$ need to be estimated based on the historical dataset. A natural way is to maximize the likelihood function, which is expressed as

$$L(\boldsymbol{\psi}|\boldsymbol{D}) = \prod_{i=1}^{I} \int p(F_i, \delta_i|\boldsymbol{B}_i; \boldsymbol{\psi}) p(\boldsymbol{Y}_i|\boldsymbol{B}_i; \boldsymbol{\psi}) \pi(\boldsymbol{B}_i; \boldsymbol{\psi}) d\boldsymbol{B}_i \,, \tag{8}$$

where the three components in Equation (8) represent the survival model, degradation model, and distribution of $\boldsymbol{B}_{i,j}$, respectively. The expressions for each component are:

$$\begin{aligned} p\left(F_i, \delta_i \middle| \boldsymbol{B}_{i,j}; \boldsymbol{\psi}\right) &= h_i(F_i)^{\delta_i} S_i(F_i) \\ &= \left\{ h_0(F_i) \, exp\left[\sum_{k=1}^{K} \beta_k \left(A\left(\alpha_{0k} + \sum_{j=1}^{J} \alpha_{jk} r_{i,j}(F_i) \right) \right) \right] \right\}^{\delta_i} \\ &\quad \times exp\left\{ - \int_0^{F_i} h_0(u) \, exp\left[\sum_{k=1}^{K} \beta_k \left(A\left(\alpha_{0k} + \sum_{j=1}^{J} \alpha_{jk} r_{i,j}(u) \right) \right) \right] du \right\}, \end{aligned} \tag{9}$$

$$p(\boldsymbol{Y}_i|\boldsymbol{B}_i; \boldsymbol{\psi}) = \prod_{j=1}^{J} p\left(\boldsymbol{Y}_{i,j} \middle| \boldsymbol{B}_{i,j}; \boldsymbol{\psi}\right) = \prod_{j=1}^{J} \left(2\pi\sigma_j^2\right)^{-\frac{N_i}{2}} e^{\frac{-\left\|\boldsymbol{Y}_{i,j} - \boldsymbol{X}_{i,j}\boldsymbol{B}_{i,j}\right\|^2}{2\sigma_j^2}}, \tag{10}$$

$$\begin{aligned} \pi(\boldsymbol{B}_i; \boldsymbol{\psi}) &= \prod_{j=1}^{J} \pi\left(\boldsymbol{B}_{i,j} \middle| \boldsymbol{\psi}\right) \\ &= \prod_{j=1}^{J} \frac{1}{(2\pi)^{\frac{q_j+1}{2}} \left|\boldsymbol{\Sigma}_0^{(j)}\right|^{\frac{1}{2}}} exp\left(- \frac{\left(\boldsymbol{B}_{i,j} - \boldsymbol{\mu}_0^{(j)}\right)^T (\boldsymbol{\Sigma}_0^{(j)})^{-1} \left(\boldsymbol{B}_{i,j} - \boldsymbol{\mu}_0^{(j)}\right)}{2} \right). \end{aligned} \tag{11}$$

For the complex likelihood equation without a closed-form expression, numerical integration and optimization techniques such as the Expectation-Maximization algorithm are frequently used in practice (Wulfsohn and Tsiatis 1997). However, due to high-dimensional integration, the process of convergence might be very slow.

To address this issue, we adopt a commonly used alternative, namely the two-stage estimation method (Tsiatis, Degruttola, and Wulfsohn 1995). Specifically, at the first step, the degradation signal of sensor *j*

from unit i is fitted through a mixed-effects model inference procedure. At the second step, the estimated measurements of $r_{i,j}(t) = \boldsymbol{X}_{i,j,t}\boldsymbol{B}_{i,j}$ are treated as given observations, based on which the parameters $\{h_0(t), \alpha_{jk}, \beta_k\}$ of the NN-based Cox model are trained. Although some bias may be introduced by the use of two-stage estimation method, the bias is known to be negligible (Wulfsohn and Tsiatis 1997). Using two-stage method facilitates easier implementation of our model to engineering practice with negligible (if any) bias. Therefore, we believe two-stage method is a viable option for our model. Below, we give a detailed description of how to estimate the necessary parameters in our model.

The parameters $\left\{\boldsymbol{\mu}_0^{(j)}, \boldsymbol{\Sigma}_0^{(j)}, \sigma_j^2\right\}$ can be easily obtained based on the restricted maximum likelihood estimation. The parameters estimation in the Cox model shown in Equation (9) are estimated by maximizing the Cox partial likelihood (Cox 1972). Then, the non-parametric baseline hazard is estimated based on the partial likelihood estimation results. The partial likelihood can be expressed as

$$L\big(\{\alpha_{jk}, \beta_k, j = 0,1, \dots, J, k = 1, \dots, K\}\big) = \prod_{i=1}^{N}\left(\frac{e^{w_i(F_i)}}{\sum_{\tau\in\boldsymbol{\Theta}(F_i)} e^{w_\tau(F_i)}}\right)^{\delta_i}, \tag{12}$$

where $w_i(F_i)$ is defined in (6) and $\boldsymbol{\Theta}(F_i)$ is the risk set given as

$$\boldsymbol{\Theta}(F_i) = \{j | F_j \geq F_i\}. \tag{13}$$

In (12), it is assumed that only one failure occurs at each distinct event time. In practice, however, multiple events may occur at the same time, i.e., tied events (Hertz-Picciotto and Rockhill 1997). If only a single failure is counted at each event time, it is possible to get underestimated results; hence, we adopt the Efron approximation to handle the issue associated with tied events (Efron 1977). Let $\tau_1, \dots, \tau_M$ be the M ordered and distinct failure times, then the likelihood function can be expressed as:

$$\begin{aligned} &L\big(\alpha_{jk}, \beta_k, j = 0,1, \dots, J, k = 1, \dots, K\big) \\ &\qquad = \prod_{m=1}^{M}\left(\frac{\prod_{\varphi\in\boldsymbol{H}(\tau_m)} e^{w_\varphi(\tau_m)}}{\prod_{l=1}^{d_m}\left(\sum_{i\in\boldsymbol{\Theta}(\tau_m)} e^{w_i(\tau_m)} - \frac{l-1}{d_m}\sum_{\varphi\in\boldsymbol{H}(\tau_m)} e^{w_\varphi(\tau_m)}\right)}\right), \end{aligned} \tag{14}$$

where d_m is the number of failures at time τ_m. $\boldsymbol{H}(\tau_m)$ is a set of units that failed at time τ_m.

To take advantage of the learning ability of NN, the objective function of NN can be directly derived from the negative log-likelihood function of Equation (14). Then, the optimal values of $\{\alpha_{jk}, \beta_k, j =$

$0,1,\dots,J, k=1,\dots,K\}$ can be obtained by minimizing the loss function using the back-propagation with gradient descent algorithm (Huang *et al.* 2007).

Due to a large number of parameters, overfitting problem often arises for NNs (Mahamad, Saon, and Hiyama 2010). Therefore, we impose a penalty to the loss function to encourage $w_i(t)$ to not deviate too far from zero as below (Kvamme, Borgan, and Scheel 2019):

$$Loss = -\sum_{m=1}^{M}\left\{\sum_{\varphi\epsilon \boldsymbol{H}(\tau_m)} w_\varphi(\tau_m) - \sum_{l=1}^{d_m} log\left(\sum_{i\in \boldsymbol{\Theta}(\tau_m)} e^{w_i(\tau_m)} - \frac{l-1}{d_m}\sum_{\varphi\epsilon \boldsymbol{H}(\tau_m)} e^{w_\varphi(\tau_m)}\right) - \Psi\sum_{i\in \boldsymbol{\Theta}(\tau_m)}|w_i(\tau_m)|\right\} \quad (15)$$

where Ψ is a tuning parameter, which can be tuned by performing cross-validation.

Lastly, for the baseline hazard estimation, Breslow approximations to the partial log-likelihood can be used (Breslow 1972) as:

$$\hat{h}_0(\tau_m) = \frac{d_m}{\sum_{i\in \boldsymbol{\Theta}(\tau_m)} e^{\boldsymbol{w_i}(F_i)}}. \quad (16)$$

2.3 Online Bayesian Updating and RUL Prediction

Once the estimated parameters are obtained at the offline stage, the RUL prediction can be made at the online stage for an in-service unit. We denote the in-service unit as c and denote the available degradation signals collected from unit c as $y_{j,1:t^*}$, for $j = 1,2,3,\dots,J$, where t^* indicates the time instant when the prediction is to be made. To get the RUL prediction accurately, it requires updating the degradation model parameters sequentially by combining historical data and individual-specific information (Zhou *et al.* 2014). Under the Bayesian framework, the posterior distribution $p\left(\boldsymbol{B}_j \mid y_{j,1:t^*}\right)$ of sensor j for unit c is also multivariate normal with mean $\boldsymbol{\mu}_{t^*}^{(j)}$ and variance $\boldsymbol{\Sigma}_{t^*}^{(j)}$, which can be computed as

$$\boldsymbol{\Sigma}_{t^*}^{(j)} = \left[\frac{\boldsymbol{X}_{j,1:t^*}^T \boldsymbol{X}_{j,1:t^*}}{\hat{\sigma}_j^2} + \left(\boldsymbol{\Sigma}_0^{(j)}\right)^{-1}\right]^{-1}, \quad (17)$$

$$\boldsymbol{\mu}_{t^*}^{(j)} = \boldsymbol{\Sigma}_{t^*}^{(j)}\left[\left(\boldsymbol{\Sigma}_0^{(j)}\right)^{-1}\boldsymbol{\mu}_0^{(j)} + \frac{\boldsymbol{X}_{j,1:t^*}^T\, y_{j,1:t^*}}{\hat{\sigma}_j^2}\right], \quad (18)$$

where $\boldsymbol{X}_{j,1:t^*}$ is the design matrix for the jth degradation signal up to time t^*, $\hat{\sigma}_j^2$ is the estimated variance for the measurement error of sensor j.

Given the trained population-level parameters $\{\hat{h}_0(t), \hat{\alpha}_{jk}, \hat{\beta}_k\}$ and $p(\boldsymbol{B}_j|\, y_{j,1:t^*})$, the marginal survival function of unit c can be obtained by

$$S(t|t^*, y_{j,1:t^*}) = \int S(t|t^*, \boldsymbol{B}_j) P(\boldsymbol{B}_j|y_{j,1:t^*}) d\boldsymbol{B}_j, \tag{19}$$

where $S(t|t^*, \boldsymbol{B}_j)$ is the survival function formulated as

$$S(t|t^*, \boldsymbol{B}_j) = exp\left\{-\int_{t^*}^{t} \hat{h}_0(u)\, exp\left(\sum_{k=1}^{K} \hat{\beta}_k \left(A\left(\hat{\alpha}_{0k} + \sum_{j=1}^{J} \hat{\alpha}_{jk} r_j(u)\right)\right)\right) du\right\}. \tag{20}$$

As shown in (19) and (20), these functions are analytically intractable due to the high-dimensional integration. Therefore, we use Monte Carlo simulation approach. We first generate N_s samples $\{\overline{\boldsymbol{B}}_j^{(s)}, s = 1, \dots, N_s\}$ for $\boldsymbol{B}_j$ from the updated posterior distribution $p(\boldsymbol{B}_j|\, y_{j,1:t^*})$ and, conditioning on each sample, the predicted degradation measurements at future time step $t^* + l$ is obtained as $\hat{r}_j^{(s)}(t^* + l) = \boldsymbol{X}_{t^*:t^*+\boldsymbol{l}} \overline{\boldsymbol{B}}_j^{(s)}$. Then, the estimated marginal survival function becomes

$$\hat{S}(t|t^*,\, y_{j,1:t^*}) = \frac{1}{N_s} \sum_{s=1}^{N_s} exp\left\{-\int_{t^*}^{t} \hat{h}_0(u) exp\left(\sum_{k=1}^{K} \hat{\beta}_k \left(A\left(\hat{\alpha}_{0k} + \sum_{j=1}^{J} \hat{\alpha}_{jk} \hat{r}_j^{(s)}(u)\right)\right)\right) du\right\}. \tag{21}$$

The integration in (21) is done numerically by using the Gauss–Legendre quadrature approach (Hildebrand 1987). Once the marginal survival function is obtained, the expected RUL can be computed by

$$RUL(t^*) = \int_{t^*}^{\infty} \hat{S}(t|t^*,\, y_{j,1:t^*}) dt\,. \tag{22}$$

3 Model Demonstration and Performance Evaluation

In this section, we perform two case studies to demonstrate and evaluate the proposed joint modeling framework. The first one is a Monte Carlo simulation study. Unlike the real-world applications where the underlying true degradation and failure mechanisms are unknown and unobservable, in this simulation, we know the true model generating the data. Therefore, we can provide an in-depth discussion about the characteristics of the proposed method. The second study is based on a high-fidelity turbine engine simulation software developed by the laboratory of thermal turbo machines (Mathioudakis *et al.* 2002). We can change various operating parameters of the turbine engine through the simulation software to trigger failure of the unit, then use the data for our study. Since the true underlying model of the simulation software

is vastly different from the structure of our joint modeling framework, we can evaluate the performance of the proposed method under a more realistic scenario and provide a set of practical insights.

3.1 Simulation Study

For the sake of simplicity, we assume that the degradation signals are collected from two sensors, i.e., $J = 2$. To generate dataset $\boldsymbol{D}_i = \{\boldsymbol{Y}_i, F_i, \delta_i\}$, first we generate historical data $\boldsymbol{Y}_i$ by assuming the true degradation signal path as

$$y_{i,j}(t) = \boldsymbol{X}_{i,j,t}\boldsymbol{B}_{i,j} + \varepsilon_j, \tag{23}$$

where $\boldsymbol{X}_{i,j,t} = [1, t, t^2]$ for $j \in \{1, 2\}$. The parameters used in (23) are listed in Table I. To simulate the event times, we assume that the true baseline hazard function $h_0(t)$ is a Weibull distribution as

$$h_0(t) = \lambda\gamma t^{\gamma-1}, \tag{24}$$

where λ is the shape parameter and γ is the scale parameter. They are set as $\lambda = 0.0001$ and $\gamma = 1.05$, respectively. We use a square function to model the non-linear relationship among covariates in the Cox model. The true hazard rate function of unit i is set as

$$h_i(t) = 0.0001 \times 1.05 \times t^{1.05-1} exp\left\{1.05\left((\boldsymbol{X}_{i,1,t}\boldsymbol{B}_{i,1}\right)^2 + 0.75\left(\boldsymbol{X}_{i,2,t}\boldsymbol{B}_{i,2}\right)^2\right\}. \tag{25}$$

Table I. Parameter settings for degradation signals generation

Parameters	$j = 1$	$j = 2$
$\boldsymbol{\mu}_0^{(j)}$	$[2.5, 0.01, 0.01]$	$[1.5, 0.01, 0.01]$
$\boldsymbol{\Sigma}_0^{(j)}$	$\begin{bmatrix} 0.2 & -4e-4 & 7e-5 \\ -4e-4 & 3e-6 & 1e-7 \\ 7e-5 & 1e-7 & 3e-6 \end{bmatrix}$	$\begin{bmatrix} 0.1 & -2e-5 & 4e-6 \\ -2e-5 & 3e-6 & 1e-8 \\ 4e-6 & 1e-8 & 3e-6 \end{bmatrix}$
σ_j^2	0.01	0.01

With the true hazard rate function, the failure time distribution can be easily expressed as $f_i(t) = h_i(t)S_i(t)$, and the failure time T_i for unit i can be sampled from $f_i(t)$. Figure 4 shows an example of true survival curve and true failure time distribution of randomly selected unit i. For the censoring time C_i and censoring indicator δ_i, we assume that 5% of units are censored and C_i is randomly sampled from a uniform distribution with the range from 0 to T_i. The amount of censored units may negatively affect the overall

performance of the prediction method. Although not included in this paper, according to our sensitivity study, our model still performs well even when we assume a high percentage of censored units, e.g., 50%.

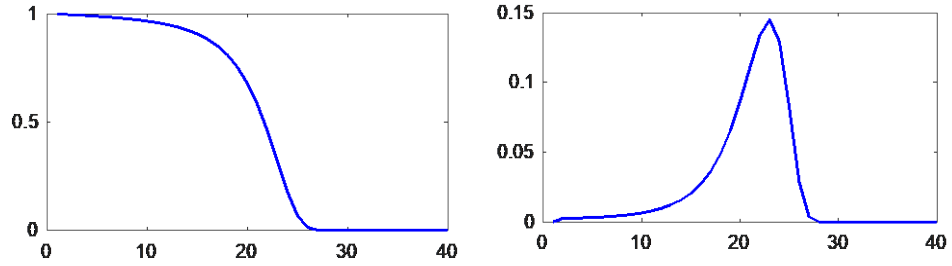

Figure 4. True survival function and failure time distribution

Table II. Description of the data generation process

(1) Generate samples $\boldsymbol{B}_{i,1}$ and $\boldsymbol{B}_{i,2}$ from normal distribution with parameters in Table I for i=1,…,N.
(2) Generate the failure times T_i for $i = 1, \dots, N$ by drawing random samples using reject sampling from $f_i(t)$ based on (5).
(3) Choose 5% of the simulated units as censored units, where the censoring time C_i is sampled from uniform distribution $U(1, T_i)$ for the selected i.
(4) Generate observations based on (23) for each unit using simulated $\boldsymbol{B}_{i,1}, \boldsymbol{B}_{i,2}$ and event time F_i, where the noise is simulated from normal distribution $\varepsilon_j \sim (0, \sigma_j^2)$ with $j = 1,2$.

A detailed stepwise description of data generation procedure is provided in Table II. The number of units N for training is set to 500. We assume the unit of time is cycle. Following the steps presented in Table II, we can obtain our training dataset $\boldsymbol{D}_i$ for $i = 1, 2, \dots, 500$. Once we have generated a historical dataset, the hyper-parameters $\boldsymbol{\psi}$ can be estimated based on the two-stage estimation method. To determine the optimal structure of NN, i.e., the number of hidden layers and number of hidden nodes, the five-fold cross validation is used on the training dataset (i.e., 20% of units are used for validation). The model with minimum validation error is eventually selected. Note that a satisfactory model fit with low loss values does not necessarily lead to a better prognostic accuracy, especially when the model is overfitted. To address this issue, in the cross validation step, we primarily focus on the RUL prediction performance instead of checking the loss values computed from the validation dataset. In other words, the optimal parameters are the ones with minimal average "prediction" error. After cross validation, we determine the structure of NN: two hidden layers with 40 neurons and 20 neurons, respectively. The tuning parameter is set as $\Psi = 0.05$.

The learning rate is set as 0.001. The sample size N_s for online prediction is 1000. To evaluate the prediction performance of the proposed method, we randomly generate degradation signals for an operating unit c using (23) with $\boldsymbol{B}_1 = [2.48, 0.0069, 0.0103]$ and $\boldsymbol{B}_2 = [1.4056, 0.0075, 0.0118]$. Then, the individualized online RUL prediction is performed on unit c.

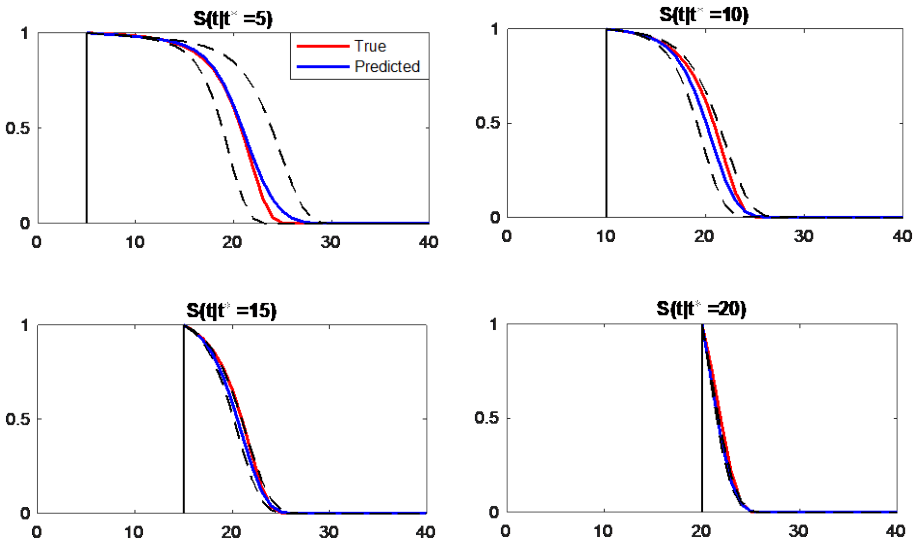

Figure 5. Comparisons between true and predicted conditional survival curves for unit c using NN-based Cox model at t^*=5, 10,15, 20. The dashed lines indicate 2.5/97.5 percentiles.

Table III. Performance evaluation of NN-based Cox model

t^*	Mean RUL			Prob. (survival in 5 cycles)			Prob. (survival in 10 cycles)		
	true	predicted	error	true	predicted	error	true	predicted	error
5	15.0797	16.1835	1.1038	0.9840	0.9807	0.0033	0.9000	0.9339	0.0339
10	8.9743	9.2295	0.2552	0.9420	0.9399	0.0021	0.4710	0.5101	0.0391
15	5.5015	5.6957	0.1943	0.6590	0.5894	0.0696	0.0000	0.0086	0.0085
20	1.7774	1.7968	0.0194	0.0043	0.0075	0.0032	0.0000	0.0000	0.0000

Figure 5 shows the comparison between true and predicted conditional survival curves at four different prediction time (t^*=5, 10, 15, 20) for unit c and the RUL prediction results are summarized in Table III along with the probabilities of survival within the next 5 and 10 cycles. To demonstrate the advantages of the proposed NN-based Cox model, we report the performance evaluation results of the standard Linear-Cox model in Figure 6 and Table IV. Because the Linear-Cox model can be considered as a special case of our model with no hidden layers, it serves as a reasonable benchmark method. From Figure 5 and Table III

at the early prediction time t^*=5, we see the largest discrepancy between the predicted values from the NN-based Cox model and the true values. The early prediction largely depends on the accuracy of the prior information estimated from the historical data because we have only a few observations from unit c. In that sense, the estimated conditional survival curve has not been individualized enough to show the unique characteristics of unit c. Rather, it primarily reflects the population's behaviors. As more observations are collected from unit c, the estimated conditional survival curve gets closer to the true survival curve; hence, the prediction performance becomes more accurate. At later prediction time (e.g., t^*=20), the estimated survival curve becomes almost identical to the true one. On the other hand, the Linear-Cox model fails to achieve satisfactory performance even at the later prediction time as shown in Figure 6 and Table IV. Due to its inadequate model structure, the Linear-Cox model is unable to capture the nonlinear relationship among degradation signals, which leads to an inaccurate prediction.

Table IV. Performance evaluation of the Linear-Cox model

t^*	Mean RUL			Prob. (survival in 5 cycles)			Prob. (survival in 10 cycles)		
	true	predicted	error	true	predicted	error	true	predicted	error
5	15.0797	15.9454	0.8657	0.9840	1.0000	0.0160	0.9000	0.9810	0.0810
10	8.9743	10.3457	1.3714	0.9420	0.9601	0.0181	0.4710	0.6399	0.1689
15	5.5015	6.0487	0.5473	0.6590	0.6942	0.0352	0.0000	0.1975	0.1975
20	1.7774	3.3423	1.5649	0.0043	0.3458	0.3415	0.0000	0.0239	0.0239

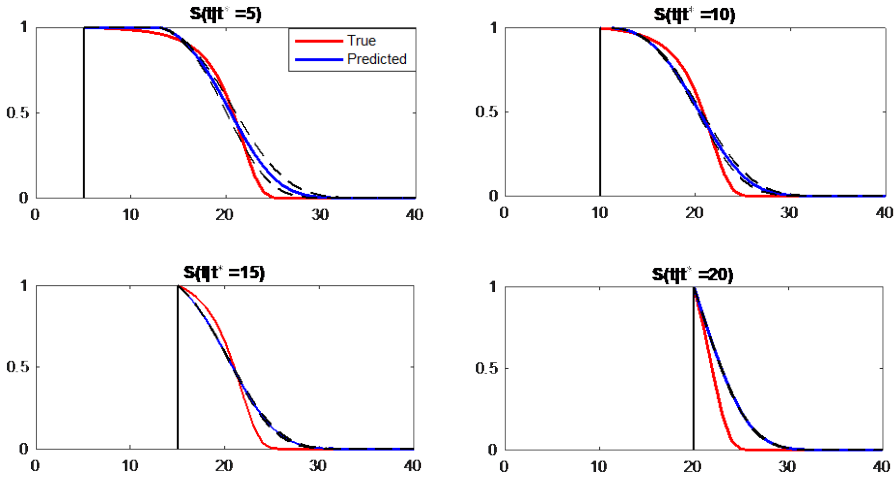

Figure 6. Comparisons between true and predicted conditional survival curves for unit c using Linear-Cox model at t^*=5, 10,15, 20. The dashed lines indicate 2.5/97.5 percentiles.

In addition to demonstrating the advantageous features of the proposed method by using an example unit c, we further evaluate the overall performance with more simulated in-service units. Specifically, we simulated 100 in-service units, i.e., $N = 100$, and computed the mean absolute error (MAE) and root mean square deviation (RMSD) defined as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{R}_i - R_i\right| \tag{26}$$

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N}\left(\hat{R}_i - R_i\right)^2}{N}} \tag{27}$$

where $\hat{R}_i$ and R_i are the predicted and true RUL of unit i, respectively.

Table V. Comparison of NN-based Cox model and Linear-Cox model at four prediction times

t^*	MAE		RMSD	
	NN-based Cox	Linear-Cox	NN-based Cox	Linear-Cox
5	1.3180	1.7375	1.6067	2.1367
10	0.5986	1.3586	0.8735	1.7843
15	0.2868	1.2210	0.3908	1.5513
20	0.1580	`1.1983	0.2167	1.4121

Table V shows the MAE and RMSD of the proposed NN-based Cox model and Linear-Cox model based on the 100 simulations. As we can see, the MAE and RMSD consistently decrease as the prediction time increases for both models, confirming our previous findings. More importantly, the proposed NN-based Cox model outperforms the Linear-Cox model at all prediction times.

3.2 Case study on Gas Turbine Engine Dataset

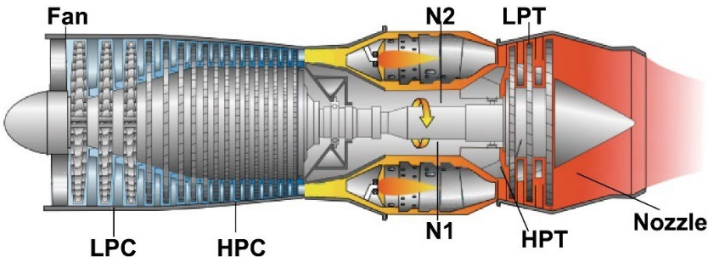

Figure 7. Simplified engine diagram simulated in the software.

To investigate the performance of the proposed method through a more practical lens, in this section, we use a high-fidelity turbine engine simulation software (Mathioudakis *et al.* 2002). The software allows users

to set different operating condition parameters to examine the influence of different operation conditions on engine performance, including flight altitude (0-15000), Mach number (0-0.90), throttle resolver angle (20-100), ambient temperature difference from ISA (-20 to 20), ram pressure recovery (0.99-1), fuel flow (0.55-1.5), low pressure shaft speed (0-4750), high pressure shaft speed (7300-13280), high pressure turbine inlet temperature (900-1400), engine pressure ratio (1.2-1.8), and thrust (15000-230000). Based on the operating parameters, the software accordingly simulates the working performance of the engine. Besides, the software also allows users to input indicative parameters of health conditions to control the degradation modes, including fan outer flow/efficiency drop, fan inner flow/efficiency drop, HP compressor flow/efficiency drop, HP turbine flow/efficiency drop, LP turbine flow/efficiency drop, and nozzle area change. By adjusting one or more of these parameters, the users can artificially introduce faults and simulate the working state of the engine with specific symptoms. As listed in Table VI, there are, in total, 16 variables in the outputs in response to 9 operation conditions and 11 health status indicator parameters that users can manipulate. Figure 7 shows a schematic diagram of a commercial aircraft gas turbine engine the simulator is designed for.

Table VI. Description of the Outputs from the Software

Symbol	***Description***	***Units***
NL	LP shaft speed	rpm
NH	HP shaft speed	rpm
P13	Fan outer outlet pressure	bar
P26	HP compressor inlet pressure	bar
T26	HP compressor inlet temperature	C
P3	HP compressor outlet pressure	bar
T3	HP compressor outlet temperature	C
T6	Exhaust gas temperature	C
EPR	Engine pressure ratio	--
T13	Fan outer outlet temperature	C
P42	HP turbine outlet pressure	bar
T42	HP turbine outlet temperature	C
P5	LP turbine outlet pressure	bar
T41	HP turbine inlet temperature	C
Thrust	Thrust	Nt
Wf	Fuel flow rate	Kg/s

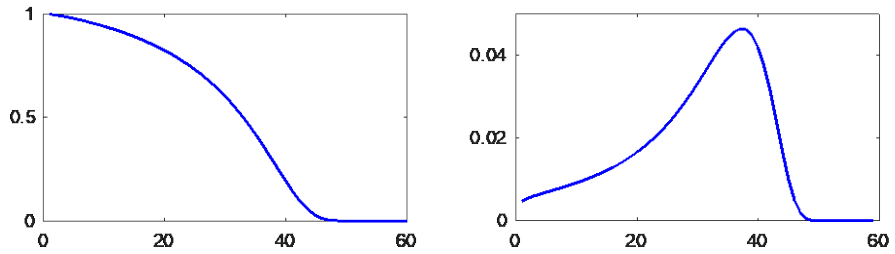

Figure 8. True survival function and failure time distribution

To simulate hard failure cases, we first model the degradation process with an exponential form and define a baseline hazard function. Then, we randomly sample failure times (hard failure) from the probability density function $f_i(t) = h_i(t)^{\delta_i} S_i(t)$. Specifically, to model the degradation processes, we select HP compressor flow drop as true indicative parameters of health conditions to control the degradation modes. The health condition of HP compressor flow drop is assumed to follow an exponential form,

$$\Gamma = 0.7\, exp(a + bt), \tag{28}$$

where a and b are assumed to follow normal distributions specified as $a \sim N(0.01, 0.005^2)$ and $b \sim N(0.045, 0.001^2)$. We set operation conditions at their default values. Based on the input health indices, the software simulates the operating process of each unit accordingly. The performance of the engine is calculated and represented by outputs of multiple degradation signals. To make it more realistic, these measurements are further contaminated by noises, which are simulated from normal distributions. To simulate event times, the true baseline function $h_0(t)$ still be assumed as a Weibull distribution with the shape and scale parameters $\lambda = 0.002$ and $\gamma = 1.15$, respectively. Figure 8 shows a true survival curve and true failure time distribution of randomly selected unit i. The remaining steps for data generation are identical to the process shown in Table II.

Once the historical dataset is generated, we evaluate the performance of the proposed method. Most existing researches on Gas turbine engine RUL prediction suggest that only the degradation signals with a monotonic (increasing or decreasing) trend contribute to the RUL prediction (Liu, Gebraeel, and Shi 2013; Gao, Wen, and Wu 2020). Following the literature, we selected 11 sensors that show a consistent

degradation trend for all units. The selected 11 degradation signals include NL, NH, T3, P26, P42, P3, T6, T26, T41, T42, and Thrust. For all of 11 degradation signals, a quadratic polynomial function is assumed, i.e., $q_j = 2$ for $j = 1,2,\ldots,11$. It is worthy to point out that, in practice, some sensors may not provide valuable information about the underlying health status. In that case, relying on too many non-informative sensors may have some negative impact on the performance of our method. Although it is not the focus of our study, we believe that the issue can be addressed by implementing various feature selection techniques such as correlation analysis prior to RUL prediction.

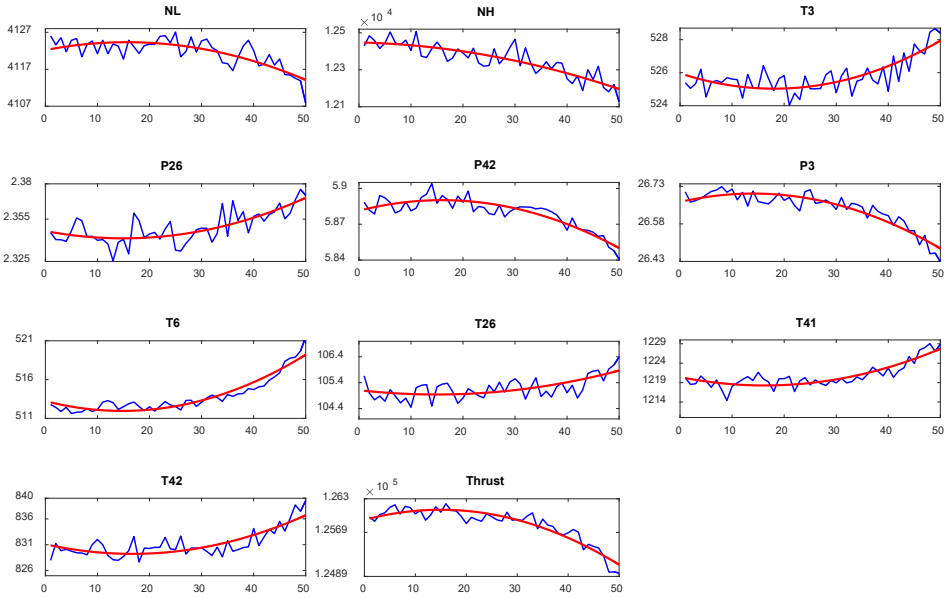

Figure 9. The raw measurements and fitted values for the selected degradation signals of unit i.

Table VII. Comparison of the true and estimated measurement errors ($\hat{\sigma}$) of the 11 degradation signals

σ	NL	NH	T3	P26	P42	P3	T6	T26	T41	T42	Thrust
True	2.00	8.00	0.50	0.0010	0.0020	0.0200	0.300	0.10	1.50	1.50	100.00
Estimated	2.01	8.60	0.51	0.0013	0.0025	0.0210	0.389	0.11	1.54	1.56	102.98

Figure 9 shows the degradation signals of a randomly selected unit and the corresponding fitted values. Table VII shows the true noise levels and estimated ones for 11 degradation signals of unit i. As we can see, the fitted degradation signals represent the true measurements with satisfactory accuracy. The fitted signals are then fed into the NN for training. The structure of NN is set to one hidden layer with 30 neurons.

The tuning parameter is set as $\Psi = 0.3$. The learning rate is 0.01 and the sample size N_s is 1,000. For demonstration purpose, Figure 10 and Table VIII summarize the results at multiple prediction times ($t^* = 10, 20, 25, 30$) for a randomly selected unit. As we discussed in the previous section, at the early stage of prediction (t^*=10), there is a noticeable difference between true and predicted conditional survival curves. However, as we collect more data from the specific in-service unit, the model becomes more tailored to the unit of our interest and the predicted condition survival curve gets closer to the true curve. Therefore, the prediction becomes much more accurate.

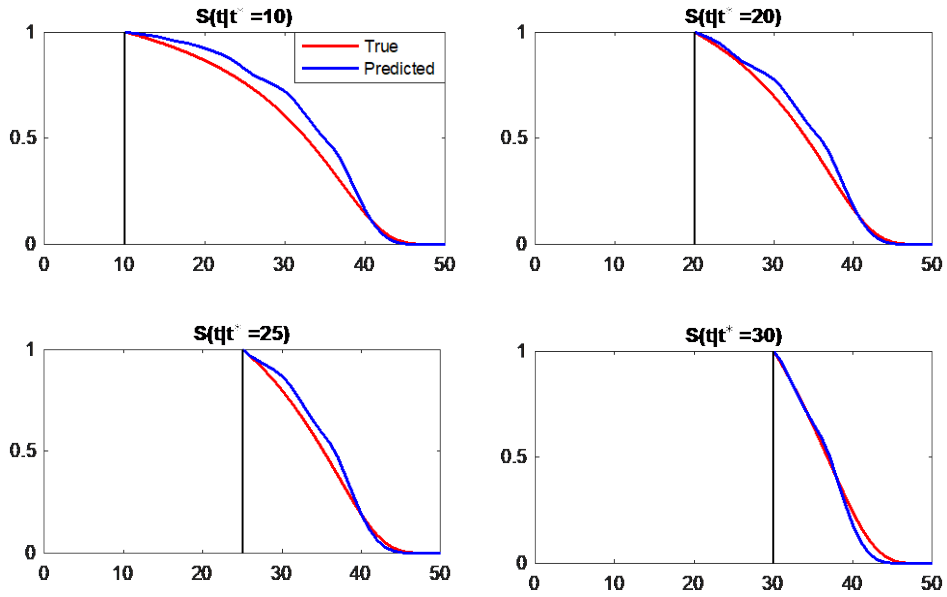

Figure 10. Comparisons between true and predicted conditional survival curves at t^*=10, 20, 25, 30.

Table VIII. Comparisons between the predicted results and the true values

t^*	Mean RUL			Prob. (survival in 5 cycles)			Prob. (survival in 10 cycles)		
	true	predicted	error	true	predicted	error	true	predicted	error
10	21.0786	23.0092	1.9306	0.8838	0.9355	0.0517	0.6415	0.7436	0.1021
20	13.4838	14.4211	0.9373	0.7404	0.8028	0.0623	0.2207	0.2635	0.0428
25	10.0117	10.6858	0.6741	0.5786	0.6453	0.0666	0.0271	0.0113	0.0159
30	6.8729	6.6060	0.2668	0.3154	0.2810	0.0343	0.0000	0.0000	0.0000

To provide a better understanding of the overall performance of the proposed method, we repeated the process 100 times each with a different simulated engine (100 test units). Also, to validate the efficacy of our method, we include more benchmark methods in addition to the Linear-Cox model shown in the

previous section. Due to the limited number of hard failure models, it is not trivial to find a model directly comparable to our method in terms of modeling capability and functionality. In the performance comparison, we include the time-dependent survival NN (TSNN), the popular long short-term memory (LSTM) network, and DeepSurv model (Katzman *et al.* 2018). The TSNN is a non-parametric one-hidden-layer neural network model (Zhang *et al.* 2019a). In this method, raw observations are used as inputs, and a fixed length of future time duration is discretized to a sequence of disjoint time intervals, say $\tau_1 < \tau_2 \cdots < \tau_K$. For each interval, an indicator 1/0 is used to represent whether the failure occurred by the time τ_k or not. In this way, the model is transformed into a combination of multiple binary classifications. To take historical data into account, an exponential function is used as a decay ratio. The LSTM network has been widely used for modeling time series data and for prediction (Zhang *et al*. 2019b); hence, it can serve as a reasonable benchmark method. The DeepSurv model is a Cox PH model with deep multi-layer perceptron. This model does not address the time-dependent influences of covariates on the time-to-event, but it is one of the most flexible Cox PH models coupled with NN. For the TSNN, the number of neurons in hidden layer is set as 40. For the LSTM, we use two LSTM layers respectively with 100 and 50 hidden neurons. The number of look-back time steps is 10. For the DeepSurv model, we use two hidden layers each with 32 hidden neurons. Table IX shows the comparison results and, based on them, we see that the proposed NN-based Cox model outperforms the benchmark methods. As all models except TSNN and DeepSurv learn more about the specific features of the in-service unit from the newly collected observations, their performance improves as we monitor the unit longer. This is highly desirable because it is more important to get an accurate prediction especially when failure is imminent. The TSNN primarily focuses on capturing the influence of degradation signals on the hazard without properly modeling the baseline hazard. The baseline hazard, in addition to the degradation signals, contributes to the overall hazard significantly. Especially in our case study, the degradation signals do not change noticeably over time. In other words, with such stable degradation signals, the baseline hazard plays a more important role in increasing the total hazard over time. Because of this nature, the TSNN fails to provide an accurate hazard estimate in this case study. The

DeepSurv model does not use any historical information from time series data. The prediction relies only on the latest observation and population-level knowledge learned from training. Therefore, for the DeepSurv model, unlike other methods, we do not see performance improvement as we acquire more observations from the specific unit. It is also noticed that the TSNN and DeepSurv model fall even behind the Linear-Cox model. This result highlights the advantages of Cox models and demonstrates the importance of adequately incorporating time-dependent covariates into the model for improving the performance of RUL prediction. As expected, the LSTM shows the worst performance because, as mentioned earlier, its primary focus is a different type of failure.

We also report the computation times for both model training and real-time prediction. Our computing resource has a 2.3 GHz Quad-Core Intel Core i7 processor and the code is written in Python. The computation time for model training is 89.9s with 1000 epochs. At the real-time prediction stage, the sample size N_s is determined by considering the trade-off between computational burden and prediction accuracy. To explain our rationale clearly, we report computation time and prediction performance for the online prediction stage with three different sample sizes, i.e., $N_s = 500, 1000, 1500$. Using a randomly chosen unit, the computation times are 5.9s, 17.4s, and 36.01s, respectively for each N_s. The computation time increases as we have a larger sample size N_s. The predictive performance on the 100 test units using these three different sample sizes is shown in Table X. We observe that larger N_s provides better prediction performance but the improvement is minimal in our case study. Therefore, considering the trade-off, we believe a small sample size such as 500 or 1000 can be used for efficient real-time prediction without sacrificing prediction accuracy. It is worth pointing out that it takes more time to train the model than to make a real-time prediction (89.9s vs. 5.9s). From Equation (15) we see that, for each epoch, the loss function needs to sum all risk sets $\Theta(F_i)$. Therefore, model training inevitably demands more computation time. We note that the heavy computational burden for model training may not be a serious issue given that the task is performed at the offline stage.

Table IX. Performance comparisons with different models

t^*	MAE					RMSD				
	NN-based Cox	Linear-Cox	TSNN	DeepSurv	LSTM	NN-based Cox	Linear-Cox	TSNN	DeepSurv	LSTM
10	1.8119	3.1082	5.7001	6.6157	14.0242	1.8706	3.1427	5.7191	9.8323	14.4009
20	0.7980	1.7276	1.8968	7.6776	8.4737	0.9320	1.7950	1.9565	8.2811	8.9093
25	0.5559	1.0227	5.3718	8.4250	6.5631	0.7070	1.1356	5.3928	9.9010	6.9588
30	0.5332	0.5772	8.5190	9.3571	4.8110	0.6241	0.6714	8.5308	11.3757	5.1793

Table X. Performance comparisons based on a different sample size N_s

t^*	MAE			RMSD		
	N_s=500	N_s=1000	N_s =1500	N_s=500	N_s=1000	N_s=1500
10	1.7567	1.8119	1.8168	1.8181	1.8706	1.8755
20	0.7678	0.7980	0.7991	0.9034	0.9320	0.9330
25	0.5417	0.5559	0.5592	0.6903	0.7070	0.7101
30	0.5434	0.5332	0.5350	0.6338	0.6241	0.6239

4 Conclusion

The advancement of IoT technology coupled with reliable sensory devices has provided an unprecedented opportunity for consistent and reliable health monitoring for complex engineering systems. At the same time, it brings a set of new research challenges on data analysis and decision making. In this paper, we proposed a joint prognostic modeling framework for RUL prediction of an individual in-service unit. The novelty of the proposed method comes from integrating both time-to-event data and multiple degradation signals into a unified framework. Unlike other joint modeling approaches available in the literature, the proposed NN-based Cox model is capable of modeling the nonlinear relationship between the coefficients and multiple degradation signals by the NN. The two-stage inference method is employed to help with parameter estimation and the Bayesian model updating approach is implemented to enable real-time online RUL prediction that is tailored to the in-service unit of our interest. The proposed method is tested and validated by a series of rigorous simulation studies including the one based on a high-fidelity gas turbine engine simulator.

Although we believe the proposed NN-based Cox model provides a plausible method for practical and flexible RUL prediction, our study has some limitations. First, the process of system degradation may

exhibit multiple phases during its lifecycle, which is hard to describe by the typical mixed-effects model. Thus, in some cases, a specialized model that can capture such multi-phase evolution of degradation signals might be needed. Second, we consider only a single failure mode. In practice, the operating units could be subject to multiple failure modes due to imposed external sources or internal forces/stresses. In that case, each failure mode might result in a distinct degradation path and service life. Furthermore, the system degradation may have been resulted from multiple competing failure modes. This will inevitably bring difficulties in the identification of failure modes as well as modeling heterogeneous degradation processes. We plan to investigate those issues and report the results in the future.

Notes on Contributors

Yuxin Wen received the B.S. degree in Medical Informatics and Engineering from Sichuan University, Chengdu, China, in 2011, the M.S. degree in Biomedical Engineering from Zhejiang University, Hangzhou, China, in 2014, and the Ph.D degree in Electrical and Computer Engineering from the University of Texas at El Paso (UTEP), El Paso, TX, USA, in 2020. She is currently an Assistant Professor in the Dale E. and Sarah Ann Fowler School of Engineering at Chapman University, Orange, CA, USA. Her research interests are focused on statistical modeling, prognostics, and reliability analysis.

Xinxing Guo received the B.S. degree in Engineering Mechanics at Department of Mechanics and Engineering Science, Peking University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree at the Department of Industrial Engineering and Management, Peking University, Beijing. His research interests include quality control and reliability engineering and machine learning.

Junbo Son received the B.S. degree in Industrial Systems and Information Engineering from the Korea University, Seoul, South Korea, in 2010, and the M.S. in Statistics and the Ph.D. degree in Industrial & Systems Engineering from the University of Wisconsin–Madison, Madison, WI, USA, in 2015 and 2016, respectively. He is currently an Assistant Professor in the Alfred Lerner College of Business & Economics at University of Delaware, Newark, DE, USA. His research interests include data-driven reliability engineering, medical informatics for advanced healthcare systems, and data analytics for solving various operations management problems.

Jianguo Wu received the B.S. degree in Mechanical Engineering from Tsinghua University, Beijing, China in 2009, the M.S. degree in Mechanical Engineering from Purdue University in 2011, and M.S. degree in Statistics in 2014 and Ph.D. degree in Industrial and Systems Engineering in 2015, both from University of Wisconsin-Madison. Currently, he is an Assistant Professor in the Dept. of Industrial Engineering and Management at Peking University, Beijing, China. He was an Assistant Professor at the Dept. of Industrial, Manufacturing and Systems Engineering at UTEP, TX, USA from 2015 to 2017. His research interests are mainly in quality control and reliability engineering of intelligent manufacturing and complex systems through engineering informed machine learning and advanced data analytics. He is a recipient of the STARS Award from the University of Texas Systems, Overseas Distinguished Young Scholars from China, P&G Faculty Fellowship, BOSS Award from MSEC, and several Best Paper Award/Finalists from INFORMS/IISE Annual Meeting. He is an Associate Editor of the Journal of Intelligent Manufacturing, and a member of IEEE, INFORMS, IISE, and SME.

References

Arisido, M. W., Antolini, L., Bernasconi, D. P., Valsecchi, M. G., and Rebora, P. (2019) Joint model robustness compared with the time-varying covariate Cox model to evaluate the association between a longitudinal marker and a time-to-event endpoint. *BMC Medical Research Methodology*, 19 (1), 1-13.

Breslow, N. E. (1972) Discussion of the paper by DR Cox. *Journal of the Royal Statistical Society, Series B*, 34, 216-217.

Chehade, A., Bonk, S., and Liu, K. (2017) Sensory-based failure threshold estimation for remaining useful life prediction. *IEEE Transactions on Reliability*, 66 (3), 939-949.

Chen, N., and Tsui, K. L. (2013) Condition monitoring and remaining useful life prediction using degradation signals: Revisited. *IIE Transactions*, 45 (9), 939-952.

Ching, T., Zhu, X., and Garmire, L. X. (2018) Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. *PLoS Computational Biology*, 14 (4), e1006076.

Cox, D. R. (1972) Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34 (2), 187-202.

Dupuy, J.-f., and Mesbah, M. (2002) Joint modeling of event time and nonignorable missing longitudinal data. *Lifetime Data Analysis*, 8 (2), 99-115.

Efron, B. (1977) The efficiency of Cox's likelihood function for censored data. *Journal of the American Statistical Association*, 72 (359), 557-565.

Faraggi, D., and Simon, R. (1995) A neural network model for survival data. *Statistics in medicine*, 14 (1), 73-82.

Gao, Y., Wen, Y., and Wu, J. (2020) A neural network-based joint prognostic model for data fusion and remaining useful life prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 32 (1), 117-127.

Gebraeel, N. Z., Lawley, M. A., Li, R., and Ryan, J. K. (2005) Residual-life distributions from component degradation signals: A Bayesian approach. *IIE Transactions*, 37 (6), 543-557.

Heng, A., Tan, A. C., Mathew, J., Montgomery, N., Banjevic, D., and Jardine, A. K. (2009) Intelligent condition-based prediction of machinery reliability. *Mechanical Systems and Signal Processing*, 23 (5), 1600-1614.

Hertz-Picciotto, I., and Rockhill, B. (1997) Validity and efficiency of approximation methods for tied survival times in Cox regression. *Biometrics*, 1151-1156.

Hildebrand, F. B. (1987), *Introduction to numerical analysis*: Courier Corporation.
Huang, R., Xi, L., Li, X., Liu, C. R., Qiu, H., and Lee, J. (2007) Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical Systems and Signal Processing*, 21 (1), 193-207.
Jardine, A. K., Lin, D., and Banjevic, D. (2006) A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20 (7), 1483-1510.
Kalbfleisch, J. D., and Prentice, R. L. (2011), *The statistical analysis of failure time data* (Vol. 360): John Wiley & Sons.
Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. (2018) DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18 (1), 24.
Kvamme, H., Borgan, Ø., and Scheel, I. (2019) Time-to-Event Prediction with Neural Networks and Cox Regression. *Journal of Machine Learning Research*, 20 (129), 1-30.
Lawless, J. F. (2011), *Statistical models and methods for lifetime data* (Vol. 362): John Wiley & Sons.
Liang, Z., Jun, Y., Haifeng, L., and Zhenglian, S. (2011) Reliability assessment based on multivariate degradation measures and competing failure analysis. *Modern Applied Science*, 5 (6), 232.
Liao, H., Zhao, W., and Guo, H. 2006. Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model. In *RAMS'06. Annual Reliability and Maintainability Symposium, 2006.*: IEEE.
Liu, K., Gebraeel, N. Z., and Shi, J. (2013) A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Transactions on Automation Science and Engineering*, 10 (3), 652-664.
Liu, X., Li, J., Al-Khalifa, K. N., Hamouda, A. S., Coit, D. W., and Elsayed, E. A. (2013) Condition-based maintenance for continuously monitored degrading systems with multiple failure modes. *IIE Transactions*, 45 (4), 422-435.
Lu, C. J., and Meeker, W. O. (1993) Using degradation measures to estimate a time-to-failure distribution. *Technometrics*, 35 (2), 161-174.
Mahamad, A. K., Saon, S., and Hiyama, T. (2010) Predicting remaining useful life of rotating machinery based artificial neural network. *Computers & Mathematics with Applications*, 60 (4), 1078-1087.
Man, J., and Zhou, Q. (2018) Prediction of hard failures with stochastic degradation signals using Wiener process and proportional hazards model. *Computers & Industrial Engineering*, 125, 480-489.
Mathioudakis, K., Stamatis, A., Tsalavoutas, A., and Aretakis, N. (2002) Computer models for education on performance monitoring and diagnostics of gas turbines. *International Journal of Mechanical Engineering Education*, 30 (3), 204-218.
McPherson, J. W. (2018), *Reliability physics and engineering: time-to-failure modeling*: Springer.
Meeker, W. Q., and Escobar, L. A. (2014), *Statistical Methods for Reliability Data*: John Wiley & Sons.
Pecht, M. G. (2010) A prognostics and health management roadmap for information and electronics-rich systems. *IEICE ESS Fundamentals Review*, 3 (4), 4_25-24_32.
Peng, W., Li, Y.-F., Yang, Y.-J., Huang, H.-Z., and Zuo, M. J. (2014) Inverse Gaussian process models for degradation analysis: A Bayesian perspective. *Reliability Engineering & System Safety*, 130, 175-189.
Prentice, R. L. (1982) Covariate measurement errors and parameter estimation in a failure time regression model. *Biometrika*, 69 (2), 331-342.
Si, X.-S., Wang, W., Hu, C.-H., and Zhou, D.-H. (2011) Remaining useful life estimation–a review on the statistical data driven approaches. *European Journal of Operational Research*, 213 (1), 1-14.
Son, J., Zhou, Q., Zhou, S., Mao, X., and Salman, M. (2013) Evaluation and comparison of mixed effects model based prognosis for hard failure. *IEEE Transactions on Reliability*, 62 (2), 379-394.
Therneau, T., Crowson, C., and Atkinson, E. (2013) Using time dependent covariates and time dependent coefficients in the cox model. *Red*, 2 (1).
Tsiatis, A. A., and Davidian, M. (2001) A semiparametric estimator for the proportional hazards model with

longitudinal covariates measured with error. *Biometrika*, 88 (2), 447-458.
Tsiatis, A. A., and Davidian, M. (2004) Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica*, 809-834.
Tsiatis, A. A., Degruttola, V., and Wulfsohn, M. S. (1995) Modeling the relationship of survival to longitudinal data measured with error. Applications to survival and CD4 counts in patients with AIDS. *Journal of the American Statistical Association*, 90 (429), 27-37.
Wang, P., and Coit, D. W. (2007) Reliability and degradation modeling with random or uncertain failure threshold. In *2007 Annual Reliability and Maintainability Symposium*: IEEE.
Wen, Y., Wu, J., Das, D., and Tseng, T.-L. B. (2018a) Degradation modeling and RUL prediction using Wiener process subject to multiple change points and unit heterogeneity. *Reliability Engineering & System Safety*, 176, 113-124.
Wen, Y., Wu, J., and Yuan, Y. (2017) Multiple-phase modeling of degradation signal for condition monitoring and remaining useful life prediction. *IEEE Transactions on Reliability*, 66 (3), 924-938.
Wen, Y., Wu, J., Zhou, Q., and Tseng, T.-L. (2018b) Multiple-Change-Point Modeling and Exact Bayesian Inference of Degradation Signal for Prognostic Improvement. *IEEE Transactions on Automation Science and Engineering*, (99), 1-16.
Wulfsohn, M. S., and Tsiatis, A. A. (1997) A joint model for survival and longitudinal data measured with error. *Biometrics*, 330-339.
Yan, F., Lin, X., Li, R., and Huang, X. (2018) Functional principal components analysis on moving time windows of longitudinal data: dynamic prediction of times to event. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67 (4), 961-978.
Ye, Z.-S., Xie, M., Tang, L.-C., and Chen, N. (2014) Semiparametric estimation of gamma processes for deteriorating products. *Technometrics*, 56 (4), 504-513.
Yousefi, S., Amrollahi, F., Amgad, M., Dong, C., Lewis, J. E., Song, C., Gutman, D. A., Halani, S. H., Vega, J. E. V., and Brat, D. J. (2017) Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific reports*, 7 (1), 1-11.
Yu, I.-T., and Fuh, C.-D. (2010) Estimation of time to hard failure distributions using a three-stage method. *IEEE Transactions on Reliability*, 59 (2), 405-412.
Yu, M., Taylor, J. M. G., and Sandler, H. M. (2008) Individual prediction in prostate cancer studies using a joint longitudinal survival–cure model. *Journal of the American Statistical Association*, 103 (481), 178-187.
Yue, X., and Kontar, R. A. (2020) Joint Models for Event Prediction from Time Series and Survival Data. *Technometrics*, 1-10.
Zhang, J., Wang, S., Chen, L., Guo, G., Chen, R., and Vanasse, A. (2019a) Time-Dependent Survival Neural Network for Remaining Useful Life Prediction. In *PAKDD (1)*.
Zhang, L., Lin, J., Liu, B., Zhang, Z., Yan, X., and Wei, M. (2019b) A review on deep learning applications in prognostics and health management. *IEEE Access*, 7, 162415-162438.
Zhou, Q., Son, J., Zhou, S., Mao, X., and Salman, M. (2014) Remaining useful life prediction of individual units subject to hard failure. *IIE Transactions*, 46 (10), 1017-1030.
Zio, E., and Di Maio, F. (2010) A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engineering & System Safety*, 95 (1), 49-57.