Explainable Prognostics and Data-driven Modeling of Complex Data in Smart and Connected Systems

By
Ye Kwon Huh

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Industrial and Systems Engineering)

at the
UNIVERSITY OF WISCONSIN-MADISON
2025

Date of final oral examination: 05/22/2025
Final Examination Committee:

Dr. Kaibo Liu (Committee Chair), Professor, Industrial and Systems Engineering
Dr. Dharmaraj Veeramani, Professor, Industrial and Systems Engineering
Dr. Shiyu Zhou, Professor, Industrial and Systems Engineering
Dr. Yongfeng Zhang, Assistant Professor, Nuclear Engineering and Engineering Physics

Acknowledgement

I would like to express my heartfelt gratitude to my academic advisor, Dr. Kaibo Liu, for his unwavering support and invaluable guidance throughout my doctoral journey. Without his feedback, deep expertise, and strong commitment to student growth and success, I would not have been able to progress this far in my academic career. He has always been a fair and approachable advisor, offering numerous opportunities for his students to grow and excel. I feel incredibly fortunate to have had him as my advisor.

I sincerely appreciate the committee members, Dr. Shiyu Zhou, Dr. Dharmaraj Veeramani, and Dr. Yongfeng Zhang, for their constructive suggestions that greatly improved the quality of this dissertation. I am also deeply grateful for their immense support, encouragement, and advice throughout my Ph.D. studies and job search.

I would like to extend my heartfelt gratitude to all the members of the Lab for System Informatics and Data Analytics. Studying alongside such a talented and dedicated group of researchers has been a privilege. I am especially thankful to Dr. Minhee Kim, as she was a great student mentor during the most challenging, early stages of my Ph.D. journey. In addition, I would like to thank my dear friends and classmates that helped me throughout my Ph.D. I hope nothing but the best in their future endeavors.

Last but certainly not least, I am most thankful for the support from my family. I thank my father Wansuk Huh and my mother Seo Jung Im for everything. Their unconditional love and encouragement have been my greatest source of strength. I owe all my achievements and accomplishments to them.

Table of Contents

List of Tables

List of Figures

Abstract

Accurate prognostics is crucial for improving the reliability and functionality of modern engineering systems. To develop accurate and reliable prognostic models, it is essential to gather and analyze sensor signal information. Fortunately, recent advances in sensor and integrated circuit technology have made it easier than ever to install, collect, and process vast amounts of sensor data. These technological advancements have brought novel developments in degradation modeling and prognostics of many smart and connected systems. However, despite these advancements, there remain four major challenges that must be addressed to ensure reliable performance in many complex real-life scenarios:

- Alignment with prior domain knowledge: how to guarantee that the prognostic model aligns with prior domain knowledge of the degradation process.
- Accurate and reliable prognostics with uncertainty quantification: how to obtain remaining useful life (RUL) predictions that are both accurate and reliable. Specifically, how to assess the "confidence" of the RUL predictions.
- Explainable prognostics: instead of a black-box model, how to obtain explainable insights into the underlying system's status and degradation dynamics. For instance, one might be interested on how to identify the most "informative" sensors that significantly affect the degradation process.
- Handling multi-type data: how to draw prognostic insights from different data types like longitudinal sensor data and discrete event data.

This dissertation focuses on explainable prognostics and data-driven modeling of complex data. Specifically, it investigates various statistical and machine learning techniques for deriving critical insights of the degradation status and RUL prediction of smart and connected systems. The novel

methodologies discussed in this work allow: (i) precise alignment of data-driven, prognostic models with prior domain knowledge; (ii) accurate and reliable RUL predictions with uncertainty quantifications; (iii) explainable insights into complex systems with high-dimensional multivariate sensor data by identifying the informative sensors; (iv) prognostic insights from both continuous sensor signal data and discrete event log data; (v) fusion of multivariate sensor signal to track the underlying degradation status.

The first chapter discusses the background and current challenges with degradation modeling and prognostics in smart and connected systems, while also outlining the key objectives of this dissertation. Chapter 2 then focuses on the challenge of aligning prior domain knowledge with data-driven degradation models. Specifically, this chapter focuses on modeling a nuclear engineering-specific degradation process called void swelling. To effectively integrate prior domain knowledge on void swelling with prognostics, this chapter proposes a Bayesian hierarchical piecewise linear model that encodes prior knowledge of void swelling. Specifically, the piecewise structure effectively captures the two-stage nature of void swelling processes, while the hierarchical Bayesian component allows one to easily incorporate domain knowledge via the prior distribution. Chapter 3 discusses the challenge of obtaining high quality uncertainty quantifications when analyzing longitudinal signal data alongside time-to-event data. Due to the complex data types, it is difficult to capture the modeling uncertainties of both data types into the final RUL predictions. To overcome this challenge, this chapter proposes an integrated uncertainty quantification (IUQ) model that accurately propagates and quantifies the uncertainties from both data types. The obtained uncertainties can then be used to assess the reliability of the RUL predictions. Chapter 4 then introduces a Bayesian spike-and-slab sensor selection approach for high-dimensional prognostics. Many existing sensor selection methods struggle to select

informative sensors in high-dimensional scenarios, where there are more sensors relative to the number of training units. On the contrary, the proposed method boasts superior sensor selection performance in high-dimensional scenarios. The main motivation of this work is based on a Bayesian spike-and-slab prior imposed on the sensor fusion coefficients. Imposing this prior allows the model to produce sparse solutions that prioritize information from informative sensors. The informative sensors are then simultaneously fused into a 1-D health index to better characterize the degradation process. Chapter 5 presents an uncertainty-informed neural network-based prognostic model for multi-type data. The main contribution of this proposed method is that it extracts prognostic insights from both continuous signal data and discrete event data. The proposed model has sub-models designed for each data type, which are then jointly trained to minimize any bias in RUL prediction. One challenge with joint training is that the model can easily fall into a local extremum due to the complex data types and model structures. To overcome this challenge, the proposed method leverages task-specific uncertainty information to automatically weigh the loss functions. This allows the network to automatically balance the loss function and prevent the model from over/underfitting. Finally, Chapter 6 includes a summary of the main contributions as well as future research directions.

In summary, the following dissertation focuses on developing reliable and explainable degradation modeling and prognostic analysis methodologies for smart and connected systems. The proposed works offer substantial potential for improving efficiency, reliability, and functionality in many applications including manufacturing, energy systems, healthcare and general Internet of Things (IoT) systems.

Chapter 1 Introduction

1.1 Motivation and Overview

All systems eventually degrade over time and experience failure. Knowing the failure time, typically defined as the time when the degradation status reaches a predefined threshold or cannot perform its normal operations, is critical for improving the reliability and functionality of the system. For instance, one can predict the remaining useful life (RUL) of the system and conduct preventative maintenance decisions when the unit is close to the end of its life. Practitioners can ensure reliable operations and avoid unnecessary downtime caused by reactionary maintenance. However, one fundamental challenge is that the underlying degradation status is unobservable and needs to be inferred. To infer the unobserved underlying degradation status, a common approach is to monitor and analyze the sensor signals. For instance, as the fan of an aircraft turbofan engine degrades, the physical speed of the fan tends to decrease over time. Therefore, modeling and monitoring the fan's physical speed allows one to predict the failure time of the turbofan engine.

In recent years, there has been a plethora of literature on leveraging sensor signals for degradation modeling and RUL prediction. These approaches typically assume that the underlying degradation status can be characterized using a univariate sensor signal [1] or multivariate sensor signals [2]. Many popular models have been developed under this assumption, including statistical models, machine learning models, and deep learning models. A comprehensive review of existing degradation modeling approaches will be provided in Chapter 2.1.1.

One major paradigm shift in modern engineering systems has been driven by innovations in modern sensing and integrated circuit technology. These innovations have enabled Internet of Things (IoT) systems to autonomously gather degradation (i.e., sensor) signals, process

information at the edge, and make informed decisions in remote environments. This revolution has spurred the rapid integration of IoT systems across various sectors like manufacturing, healthcare, and energy systems. While this revolution provides new opportunities for both researchers and practitioners, it also introduces new challenges for effective and reliable degradation modeling and prognostics.

The first challenge in modern data-driven degradation models is ensuring that they are properly aligned with prior domain knowledge. Direct application of purely data-driven, black-box models like neural networks without considering the underlying degradation dynamics can result in erroneous predictions that contradict physical laws or known process behaviors. For instance, degradation processes typically exhibit monotonic behavior, as degradation is an irreversible process without any maintenance [3]. Failing to account for this property can lead to incorrect implications about the degradation trends and inaccurate RUL predictions. To fully harness the power of data-driven models, it is essential to carefully incorporate domain knowledge in the model design stage.

Second, it is crucial to assess not only the accuracy of RUL predictions, but also their reliability/uncertainty. Degradation processes are stochastic by nature due to the multiple sources of uncertainty stemming from measurement errors and unit-to-unit variability. Therefore, it is crucial to provide accurate uncertainty quantifications alongside the RUL predictions. These uncertainty quantifications can then be used for subsequent maintenance decisions and risk analysis. However, obtaining accurate uncertainty quantifications is nontrivial as modern engineering systems collect diverse data types ranging from time-to-event data and continuous sensor signal data. Therefore, there is a strong need for a systematic procedure for obtaining

accurate and reliable uncertainty estimates that effectively capture the modeling uncertainty from each data type and integrate them into the final RUL predictions.

Third, recent developments in sensor technology have led to the widespread use of sensors to monitor and analyze system status. These sensors capture different facets of the system and the underlying degradation process. A unique and longstanding challenge of analyzing such multivariate sensor signals is that each sensor has varying degrees of relevance to the underlying degradation process. It is possible that some sensors are "informative" and provide strong insights on the degradation status, while some sensors are "uninformative" and do not provide such insights [4]. This sensor selection challenge has become increasingly difficult in modern engineering systems, where technological advances have made it practical to adopt numerous sensors. As a result, sensor signals collected from these modern systems are frequently high-dimensional, with the number of sensors being similar or much larger than the number of available training units [5]. Therefore, how to effectively identify informative sensors in modern, high-dimensional systems is highly desirable, as the informative sensors can provide interpretable insights of the degradation process.

Fourth, a critical limitation of existing methods is that they struggle to simultaneously extract prognostic insights from multi-type data, specifically discrete event data and continuous signal data. One way to address this challenge is to use deep learning models, as they can effectively handle multimodal, multi-type data with relative ease. However, a key challenge is designing an effective model architecture and joint training strategies that ensure reliable prognostic performance. Since each data type captures have distinct temporal dynamics and correlations with the underlying degradation process, naïve training strategies can easily lead the model to fall in a local extremum and have suboptimal performance [6]. Therefore, it is crucial to develop effective

training strategies and loss functions that automatically evaluate the importance of each data type and associated task (i.e., classification, regression).

This dissertation aims to address the above challenges by exploring advanced statistical and machine learning methodologies for complex, multi-type data.

1.2 Objectives

The objectives of this research are:

(i) developing a novel data-driven approach for modeling a nuclear specific degradation process known as "void swelling". The data-driven approach seamlessly incorporates prior nuclear engineering knowledge to model and predict the degree of void swelling.

(ii) proposing an integrated uncertainty quantification model for joint models with time-to-event data and longitudinal signal data. This approach propagates modeling uncertainties from both data types and integrates them to the final RUL predictions.

(iii) developing a Bayesian spike-and-slab prior sensor selection approach for systems with high-dimensional sensor signals with varying levels of correlation.

(iv) establishing a deep learning-based prognostic model for extracting prognostic insights from continuous signal data and discrete event data. This approach avoids over/underfitting issues by leveraging task-specific uncertainty to weigh the joint loss function.

1.3 Outline of the Dissertation

The remainder of the dissertation is organized as follows. Chapter 2 proposes a data-driven approach for modeling the progression of void swelling. This is the first approach to integrate nuclear-specific domain knowledge with statistical modeling techniques to achieve more accurate

predictions of the future degree of void swelling. Our innovative idea is to encode domain-specific information into a Bayesian hierarchical design, which allows the model to satisfy the shape constraints of void swelling processes. Also, the information on the changepoint (i.e., when the void swelling process transitions from the transient regime to the steady-state regime) is encoded into the model design via the parameter prior distributions.

Chapter 3 focuses on obtaining accurate uncertainty quantifications when jointly modeling two different data types: time-to-event data and longitudinal signal data. Accurately tracking the uncertainty in joint models is challenging as each data type and its sub-model captures different modeling uncertainties. To overcome this challenge, we propose an integrated uncertainty quantification (IUQ) model that propagates the modeling uncertainties of both data types, which are then eventually integrated into the resulting RUL predictions. Evaluation results show that the IUQ model provides more accurate uncertainty quantifications than existing approaches, providing practitioners with a more effective way to assess the reliability of RUL predictions.

Chapter 4 delves into a sensor selection approach for high-dimensional systems. Here, high-dimensional refers to systems in which the number of sensors p is similar or higher than the number of available training units N (i.e., $N \approx p$ or $N < p$). Sensor selection in high-dimensional systems is difficult due to the low signal-to-noise ratio and curse of dimensionality. Drawing inspiration from Bayesian spike-and-slab priors, we propose a novel Bayesian sensor selection approach that selects informative sensors and then fuses them into an informative 1-D health index (HI) for further prognostic analysis. Evaluation results on many high-dimensional scenarios demonstrate the method's superior prognostic performance and ability to discern informative sensors from uninformative ones.

Chapter 5 focuses on a deep learning approach for simultaneously obtaining prognostic insights

from discrete event data and continuous signal data. The proposed network contains three predictors, one for the event data, one for the signal data, and the final predictor to obtain the RUL predictors. Since the proposed network contains three predictors with their own loss functions, it is difficult to jointly train the network without encountering fitting issues. To overcome this difficulty, the network leverages task-specific uncertainty information as weights for the loss function. The uncertainty information is treated as a learnable parameter and is automatically adjusted to reflect the significance of each task/data type in the joint training process. Results show that the proposed method exhibits superior prognostic performance compared to models that leverage only a single data type. In addition, detailed analysis shows that the uncertainty information leads to better prognostic performance and faster model convergence.

Finally, Chapter 6 summarizes the contributions of this dissertation.

Chapter 2 Degradation Modeling using Bayesian Hierarchical Piecewise Linear Models: A Case Study to Predict Void Swelling in Irradiated Materials

2.1 Motivation

2.1.1 Degradation Modeling

Engineering systems are prone to degradation and unexpected failures. Conventionally, maintenance was performed in a reactive manner, resulting in high operation costs, longer machine downtime, and lower functionality of the engineering system. Recent advances in degradation modeling and prognostics allow practitioners to predict system failures in advance and conduct preventative maintenance operations based on the remaining useful life (RUL) [7]. This results in higher profitability, reliability, and functionality of various systems.

Existing approaches to degradation modeling can be largely divided into physics-based models and data-driven models. Physics-based degradation models attempt to incorporate the physics of the failure mechanism and quantify the characteristics of the degradation process [8]. For instance, Oppenheimer and Loparo [9] developed a physics-based model that uses machine condition information in conjunction with a life model based on material crack growth laws to estimate the RUL of a shaft cracking in a rotor. These approaches tend to be component/system specific and struggle to describe the joint effect of multiple input variables (e.g., environmental conditions), especially when the number of input variables is very large. On the contrary, data-driven approaches overcome these difficulties by estimating the degradation status directly from the

available data (e.g., degradation signals) [1], [10]. For instance, Zheng et al. [11] employed a long short-term memory (LSTM) network to estimate the RUL of lithium-ion battery, while Zhong et al. [12] used an isolation forest to detect anomalies in gas paths.

The main motivation of this case study is to explore a novel application of data-driven degradation models to a nuclear-specific material degradation mechanism called void swelling. In the following subsection, we will further explain the details of void swelling, the related existing literature, and its similarities and dissimilarities to conventional degradation modeling applications.

2.1.2 Void Swelling

Void swelling is defined as a material degradation process caused by high-energy neutron irradiation under intermediate temperatures (i.e., ranging roughly between 30% and 50% of the metal's melting temperature). As materials are bombarded by high-energy neutrons, atoms are displaced from the lattice sites, which increases the material's volume. Figure 2.1 illustrates swelling observed in unfueled 20% cold-worked AISI 316 (i.e., stainless, austenitic Cr-Ni-Mo steels) open cladding tube in an EBR-II fast reactor [13]. From the figure, we observe that the cladding tube's volume increases, i.e., swells, after being exposed to irradiation. Excess void swelling can cause dimensional instability and even severe embrittlement of internal materials, leading to a critical impact on the functionality, economic operation, and safety of nuclear power plants [14]. As a result, accurate modeling, prediction, and early identification of void swelling in irradiated structural components are crucial for reliable NPP maintenance and management operations [15].

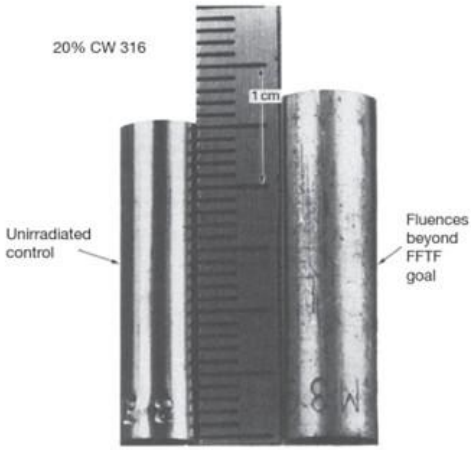
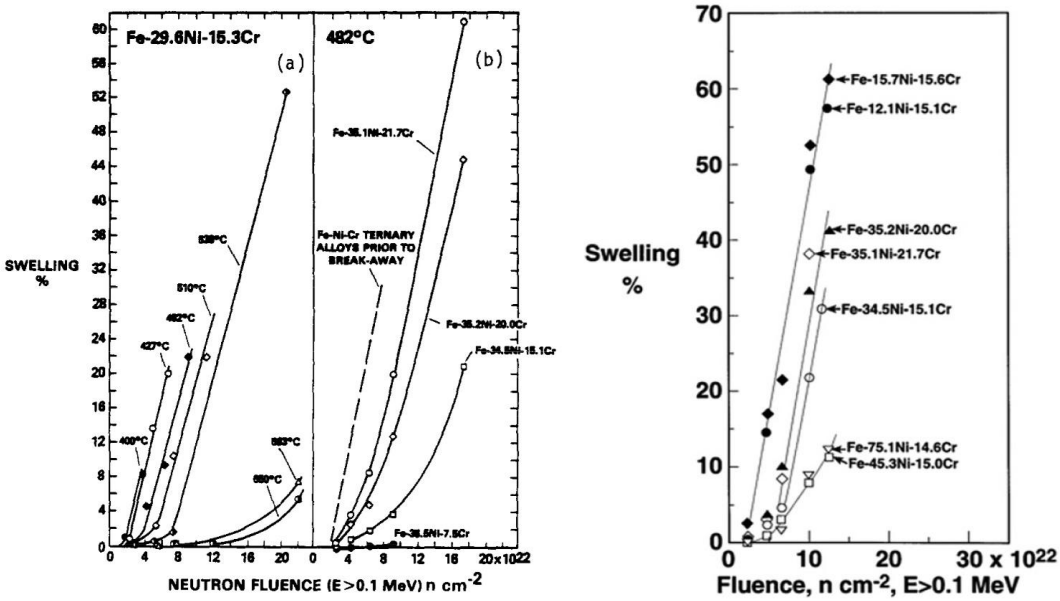


Figure 2.1 Illustration of void swelling in unfueled 20% cold-worked AISI 316 open cladding tube (Left: Before irradiation, Right: After irradiation)

Traditional approaches for understanding void swelling have been mainly based on trial-and-error, in which experiments are repeated multiple times under different settings. Hereafter, we refer to these experimental settings/factors such as alloy composition, material structure, and irradiation conditions as *covariates*. One critical limitation of such empirical approaches is that these experiments are very time-consuming and expensive as they require careful preparations, safety precautions, post-irradiation examination, and other technical considerations [16]. Another noteworthy limitation of empirical approaches is that they generally focus on how the swelling process varies with respect to a single covariate. Various works have examined the effects of a single covariate such as displacement rate [17], irradiation temperature [18], cold-work percentage [19], and irradiation type [20]. For example, Figure 2.2 shows the influence of temperature, Chromium, and Nickel content on the swelling of Fe-Cr-Ni ternary alloys in the EBR-II fast reactor [21]. However, there is still a lack of studies that analyze the joint effect of multiple covariates on the swelling process since empirical methods are too resource-intensive to repeat void swelling experiments under every possible covariate combination.



(a) (b)

Figure 2.2 (a) Influence of temperature and chromium on swelling of Fe-Cr-Ni ternary alloys irradiated in EBR-II (b) Influence of Nickel content on swelling of Fe-Cr-Ni ternary alloys in EBR-II at 427°C

Aside from empirical methods, there are general degradation models that can be applied to model void swelling. For physics-based approaches, Li et al. [22] used a Phase-field model to capture the effect of thermodynamic and kinetic properties on void nucleation and growth in irradiated materials. However, most of these methods are also covariate-specific and cannot describe the joint effect of multiple covariates on void swelling.

For data-driven degradation models, the first work that adopted a data-driven approach in the context of void swelling was by Jin, Cao, and Short [16], who applied various machine learning techniques to predict the onset of void swelling by estimating the incubation dose values (i.e., intercept values of the steady state swelling rate). However, this paper only estimated the incubation dose and not the full swelling process, and thus it only provides a restricted view of the swelling process.

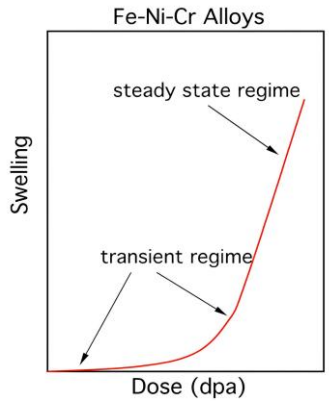


Figure 2.3 Void Swelling versus dose for Fe-Ni-Cr alloys

Unlike the previous approaches, the goal of this case study is to explicitly model and predict void swelling measurements. To the best of our knowledge, this is the first paper to directly model void swelling processes using a data-driven approach. To achieve this goal, four significant research challenges need to be addressed. First, void swelling is a function of multiple covariates with complex effects, so it is very difficult to accurately model how different covariates jointly affect the swelling process. Second, the predicted swelling trajectories must be in line with prior domain knowledge. Specifically, the trajectory of void swelling with respect to irradiation dose is divided into transient and steady state regimes as shown in Figure 2.3. The transient regime, also called the incubation state, is when either no or very small levels of swelling happen. In the subsequent steady state regime, swelling occurs at a relatively faster constant rate [21]. The predicted swelling trends must satisfy these shape constraints and clearly identify both states. Furthermore, similar to existing degradation models, void swelling is an irreversible process. Hence, the predicted void swelling trajectories should also be monotonic (i.e., nondecreasing) with respect to irradiation dose. Third, void swelling datasets are often very sparse with only one or a few measurements under a specific experimental condition (a fixed set of covariates). For example, in our case study, we have 291 unique sets of covariates with at least 1 measurement. Among the 291 covariate sets, more than 90% (i.e., 279 sets) of them have only 1 available measurement. The

inherent sparsity of the dataset often significantly compromises the accuracy of traditional data-driven approaches, which typically require a large amount of data.

Another challenge triggered by this sparsity is imbalanced data between the transient and steady-state regimes. Specifically, practical challenges during data collection often result in incomplete units that do not contain full records of the two regimes shown Figure 2.3, and instead only contain measurements of the steady-state regime or vice versa. The incomplete units can introduce unwanted bias during parameter estimation since they do not display a clear changepoint.

From these challenges, it is evident that one cannot immediately apply existing data-driven degradation models for void swelling. To address these difficulties, this article will employ various statistical techniques in Bayesian modeling and hierarchical models. In particular, we demonstrate the power of leveraging domain knowledge to design informative prior distributions used to overcome the unique challenges in void swelling modeling. With this proposed method, we hope to lay a foundation for future data-driven degradation models to better understand the latent dynamics of void swelling and similar engineering problems.

The rest of the paper is organized as follows. Section 2.2 provides a detailed problem description and research objective of this case study. Then, Section 2.2 provides a closer look at the void swelling dataset. Details of the proposed method including the hierarchical model, parameter estimation, and prediction will be discussed in Section 2.4. Then, Section 2.5 will present the numerical results, in which the effectiveness and the accuracy of the proposed method will be compared to existing benchmark methods. Next, Section 2.6 summarizes the findings and unique contributions of this case study. Finally, Section 2.7 contains the supplementary materials such as parameter settings, model adequacy checking, and recommendations for choosing the prior distributions.

2.2 Problem Description

In this study, a *unit* is defined as a collection of varying dose and swelling measurements under a specific experimental condition (a fixed set of covariates). For instance, from the tabular data in Figure 2.4, each unit has its own fixed set of covariates (e.g., unit 6 has % weight B: 0.430, % weight C: 0.6091, % weight N: 1.000, while unit 4 has % weight B: 0.48, % weight C: 0.5454, % weight N: 0.000). In addition to the covariates, each unit has one or more measurements of varying irradiation dose and corresponding void swelling %. The respective swelling curves are plotted on the right of Figure 2.4. As we can see from the plotted curves, each unit has a distinct trajectory based on its covariate values while sharing a common increasing trend. An effective model should be able to capture this unit-to-unit variability while ensuring that all predicted swelling trajectories obey the shape constraints of void swelling processes. Using this definition, our goal is to accurately predict the swelling process of a unit of interest, i.e., given its set of covariates.

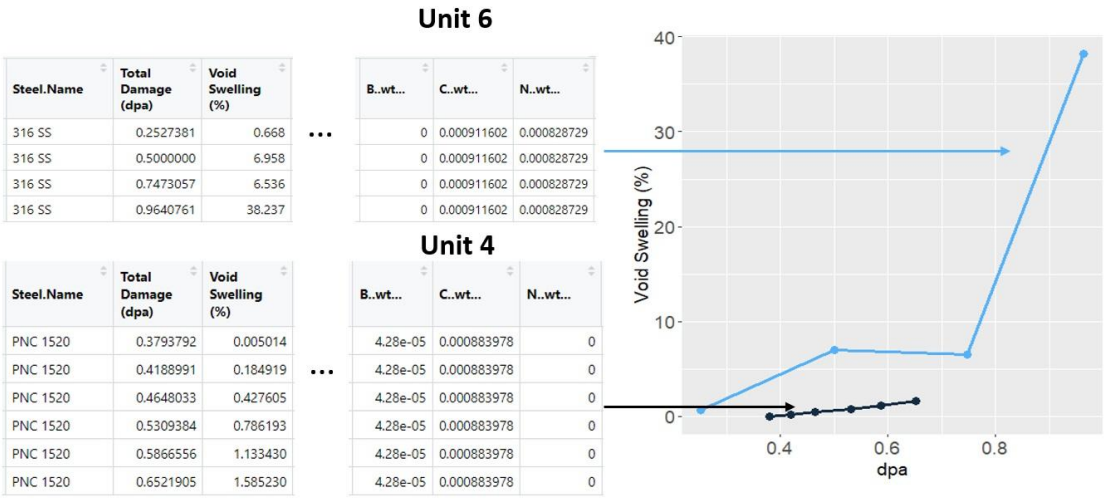


Figure 2.4 Definition of a void swelling unit.

2.3 Data Collection and Preparation

The dataset in this study is collected from publicly available literature on void swelling. There is already a tremendous number of experimental results on void swelling available through significant reviews, providing us with sufficient data. In particular, we collect measurements of dimension changes caused by void swelling (i.e., void swelling %), the associated irradiation doses (measured by displacements per atom), and the corresponding covariates (i.e., experimental parameters). Overall, we have made efforts to collect 753 measurements from 20 research papers. The datasets have been verified by the domain experts for this case study and will be made publicly available for other interested researchers in the field. To ensure that the model can accommodate swelling trends across a wide range of materials, 93 different types of steels with varying initial conditions are considered. For each measurement, we have 15 covariates ranging from irradiation temperature to alloy composition. Details of the 15 covariates can be found in Table 2.1. During data preparation, one practical difficulty is that each unit involves a different set of covariates,

Table 2.1 Data summary of covariates

Name	Mean	Standard Deviation	Units
Irradiation temperature	520.2	134.23	K
Irradiation type	Categorical Variable (5 Types): Ni6+ ion, Fe2+ ion, Neutron, Proton, Electron		
% weight Carbon	0.0492	0.0223	%
% weight Nitrogen	0.0113	0.0301	%
% weight Aluminum	0.0259	0.1559	%
% weight Silicon	0.4797	0.2465	%
% weight Phosphorus	0.0212	0.0259	%
% weight Sulfur	0.0025	0.0058	%
% weight Titanium	0.1376	0.1897	%
% weight Chromium	15.96	1.4582	%
% weight Manganese	1.246	0.7567	%
% weight Iron	51.00	24.581	%
% weight Copper	0.0053	0.0468	%
% weight Nickel	17.55	6.3192	%
% weight Molybdenum	2.050	0.9516	%

making it difficult to extract a common set of covariates across all units. To overcome this difficulty, we consider covariate-specific imputation strategies to fill in missing values. In particular, covariates regarding alloy composition (e.g., % weight Nitrogen) are imputed to 0 since it means that there are negligible levels of that element. However, the remaining covariates like irradiation type and irradiation temperature in Table 2.1 cannot be easily imputed (e.g., using mean, median) due to their unique physical properties, so they are discarded from the analysis. Next, except for irradiation type, the other covariates and the irradiation dose values are normalized to have a minimum value of 0 and a maximum value of 1. The measurements are then split into units based on the covariate values, where each unit contains at least 1 observation. Eventually, we arrive at 291 units with 395 measurements. Figure 2.5 shows the plotted swelling measurements of 12 units with more than three measurements, with the normalized irradiation dose (dpa) on the x-axis and void swelling (%) on the y-axis.

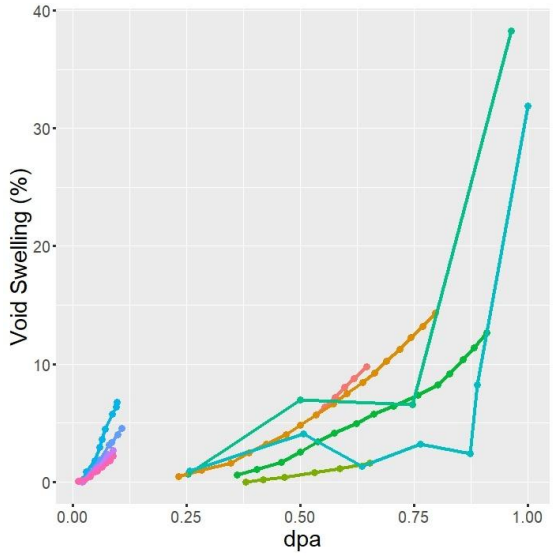


Figure 2.5 Plotted void swelling processes.

2.4 Methodology

This section contains four parts. Section 2.4.1 is a brief introduction to hierarchical regression models. In Section 2.4.2, we discuss the details of the proposed model. Next, Section 2.4.3 describes the Bayesian parameter estimation procedure. Finally, Section 2.4.4 investigates how the proposed model makes predictions on swelling evolutions.

2.4.1 Introduction to hierarchical regression models

Hierarchical regression models are one of the most widely used approaches to accommodate datasets with a nested/hierarchical structure [23]. Recently, hierarchical models have received more attention for their use in statistical process monitoring and predictive monitoring. For instance, Huberts, Schoonhoven and Does [24] used a Bayesian hierarchical model to develop a framework to monitor student performance and provide early warnings for under/overperforming students. Compared to existing linear regression-based multivariate approaches, hierarchical models can improve the estimation of process variability, make accurate predictions under limited data availability, and easily incorporate prior beliefs into the prediction stage.

A typical 2-level hierarchical regression model is formulated as follows: where equation (2.1) denotes the level 1 regression equation and equation (2.2) represents the level 2 regression equations.

$$Y_{ij} = \beta_{0i} + \beta_{1i}x_{ij} + \varepsilon_{ij}, \tag{2.1}$$

$$\beta_{0i} = \gamma_{00} + \gamma_{01}Z_i + u_{0i}, \tag{2.2}$$

$$\beta_{1i} = \gamma_{10} + \gamma_{11}Z_i + u_{1i}.$$

In the level 1 regression in equation (2.1), Y_{ij} is the jth observation from unit i; β_{0i} and β_{1i} are the intercept and slope values for unit i, respectively; x_{ij} is the corresponding level 1 predictor; and

ε_{ij} are level 1 random errors. For the level 2 regression in equation (2.2), γ_{00} and γ_{10} are the overall means of the intercept and slope across all units; Z_i is the level 2 predictor for unit i; γ_{01} and γ_{11} are the corresponding level 2 regression coefficients; and u_{0i} and u_{1i} are the level 2 random errors for β_{0i} and β_{1i}, respectively. Note that the subscript i in β_{0i} and β_{1i} indicates that the model characterizes the unit-to-unit variability by assigning varying intercept and slope values for each unit's trajectory. In summary, the level 2 predictor Z_i affects the intercept β_{0i} and slope β_{1i}. Combining these coefficients with the level 1 predictor, x_{ij}, we effectively model the observation Y_{ij}. Note that the model is not restricted to piecewise linear trends and can be extended to accommodate other trends such as linear-quadratic and quadratic-quadratic.

In the case in which a unit also has categorical covariates such as irradiation type, we use dummy variable encoding. Suppose that unit i has a categorical covariate C_i with $c \geq 1$ categories. Then, we use $c-1$ dummy variables to transform C_i into a $(c-1)$-dimensional vector $\widetilde{\boldsymbol{C}}_i = \left[C_i^{(1)}, \ldots, C_i^{(c-1)}\right]^T \in \mathbb{R}^{(c-1)\times 1}$ such that $\widetilde{\boldsymbol{C}}_i = \mathbf{0}$ if unit i belongs to the cth category and otherwise, all entries are zeros except the one corresponding to the category of unit i.

In the following subsections, we will address how the Bayesian hierarchical approach can overcome the unique challenges in void swelling processes.

2.4.2 Proposed Model

Figure 2.6 illustrates the overall framework of the proposed model. The key intuition of the proposed model is that variations in the swelling curves can be attributed to the variability in the covariates. To effectively capture this nested relationship, we consider a Bayesian hierarchical regression model in which the regression coefficients are uniquely determined by the covariates. Here, we choose to use a piecewise linear trend for the level 1 equation to capture the void swelling

trends based on the following reasons. First, the piecewise structure effectively incorporates the two-stage nature (i.e., the transient and steady states) of void swelling units. Second, the linear swelling trend in the steady state regime is in line with its definition in Section 2.1.2. Third, preliminary evaluations showed that the piecewise linear model yields smaller WAIC (Widely Available Information Criterion) [25] and prediction error than the quadratic-linear model, indicating a better fit. Hence, the level 1 equation adopts a piecewise linear model.

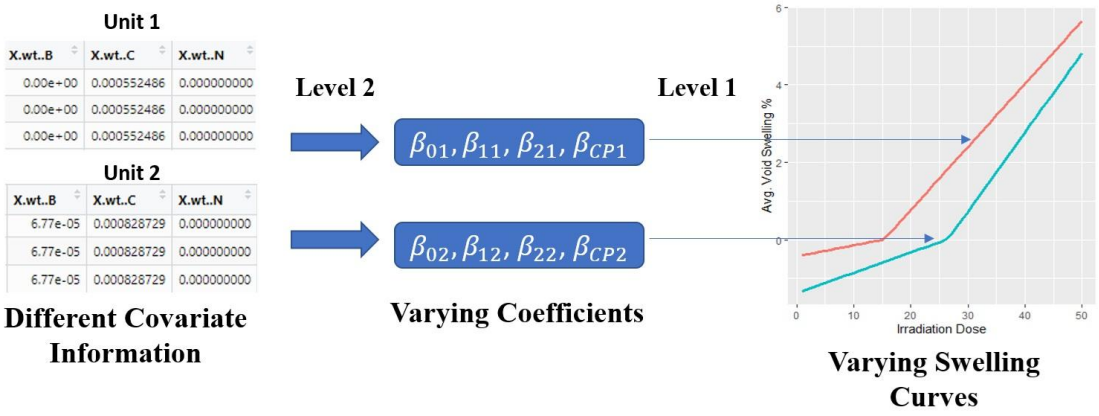


Figure 2.6 Overview of the proposed hierarchical regression framework.

Suppose there are void swelling measurements from a total of I training units, each with P covariates. As our dataset contains one categorical covariate "irradiation type" with 5 levels and 14 continuous covariates, using 4 dummy variables introduced in Section 2.4.1 yields $P = 14 + 4 = 18$ covariates for each unit. Following the notation in Section 2.4.1, the jth swelling measurement from unit i is denoted as Y_{ij} and the corresponding irradiation dose level is denoted as x_{ij}. The level 2 predictor of unit i is its covariates $\boldsymbol{Z}_i = [1, Z_{1i}, \dots, Z_{Pi}]^T \in \mathbb{R}^{(P+1)\times 1}$, with the first scalar term added for notational convenience.

$$Y_{ij} = \beta_{0i} + \beta_{1i}(x_{ij} - \beta_{CPi})\mathbb{I}\{x_{ij} < \beta_{CPi}\} + \beta_{2i}(x_{ij} - \beta_{CPi})\mathbb{I}\{x_{ij} \geq \beta_{CPi}\} + \varepsilon_{ij}, \tag{2.3}$$

$$\beta_{ai} = \gamma_{0a} + \sum_{p=1}^{P} \gamma_{pa} Z_{pi} + u_{ai} = \boldsymbol{\gamma}_a^T \boldsymbol{Z}_i + u_{0i}, a \in \{0,1,2\}, \tag{2.4}$$

$$\alpha_{CPi} = \gamma_{03} + \sum_{p=1}^{P} \gamma_{p3} Z_{pi} + u_{3i} = \boldsymbol{\gamma}_3^T \boldsymbol{Z}_i + u_{3i},$$

$$\beta_{CPi} = Inv\ Logit(\alpha_{CPi}).$$

Here, β_{CPi} represents the changepoint of unit i, the dose value where the process shifts from the transient state to the steady state. β_{0i} is the swelling value at the changepoint β_{CPi}, while β_{1i} and β_{2i} are the slopes corresponding to the transient and steady states of the piecewise linear model, and $\mathbb{I}$ is an indicator function denoting which state the process is currently at. In order to restrict the changepoint parameter β_{CPi} to be between 0 and 1, the parameter α_{CPi} undergoes an inverse logit transformation denoted by $Inv\ logit$ in equation (2.4), where $Inv\ logit(x) = \exp(x)/[1 + \exp(x)]$. ε_{ij} represents the level 1 random errors that are assumed to be normally distributed with mean 0 and variance σ^2 (i.e., $\varepsilon_{ij} \sim N(0, \sigma^2)$). As shown in equation (2.3), we use a piecewise linear trend to capture the two regimes in void swelling. In particular, if $x_{ij} < \beta_{CPi}$ (i.e., the process is in the transient state), then equation (2.3) will be $Y_{ij} = \beta_{0i} + \beta_{1i}(x_{ij} - \beta_{CPi}) + \varepsilon_{ij}$, and if $x_{ij} \geq \beta_{CPi}$ (i.e., the process is in the steady state), then $Y_{ij} = \beta_{0i} + \beta_{2i}(x_{ij} - \beta_{CPi}) + \varepsilon_{ij}$.

The level 2 equations in equation (2.4) denote the relationship between the level 1 regression coefficients $\boldsymbol{\beta}_i = [\beta_{0i}, \beta_{1i}, \beta_{2i}, \beta_{CPi}]$ and the covariates (i.e., level 2 predictors) $\boldsymbol{Z}_i$. Here, $\boldsymbol{\gamma}_0, \dots, \boldsymbol{\gamma}_3 = [\gamma_{03}, \dots, \gamma_{P3}]^T \in \mathbb{R}^{(P+1)\times 1}$ are the concatenated vectors of level 2 regression coefficients. Finally, $u_{0i}, \dots, u_{3i}$ are the level 2 random errors of $\beta_{0i}, \dots, \alpha_{CPi}$. The level 2 random errors are assumed to follow a multivariate normal distribution with mean $\mathbf{0}$ and an unknown covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{4\times 4}$ as shown in equation (2.5).

$$[u_{0i} u_{1i}, u_{2i}, u_{3i}] \sim MVN(\mathbf{0}, \boldsymbol{\Sigma}), \tag{2.5}$$

The proposed model allows unit-level variability in swelling trends through the following regression parameters: The changepoint (β_{CPi}), y-intercept at the changepoint (β_{0i}), transient

slope (β_{1i}), and steady state slope (β_{2i}). Furthermore, we impose an inequality constraint $\beta_{1i} < \beta_{2i}$, and nonnegative constraints on $\beta_{0i}, \beta_{1i}, \beta_{2i} \geq 0$ for all i to allow a faster trend during the steady state regime than the transient regime and a nondecreasing trend with positive swelling values at the changepoint, resulting in an interpretable model that is also consistent with the existing domain knowledge.

2.4.3 Bayesian Parameter Estimation

The most widely used methods for parameter estimation in hierarchical models are maximum likelihood (ML) and restricted maximum likelihood (REML) estimation [26]. These likelihood-based methods are considerably faster than Bayesian methods, but their performance suffers in terms of bias and coverage [27]. Indeed, Bayesian methods provide several advantages over likelihood-based methods at the cost of higher computation requirements. First, they quantify the uncertainties of the model parameters, which can then be used to evaluate the reliability of the parameter estimates. Second, studies have shown that estimates made by Bayesian methods are more stable and robust in small datasets than likelihood-based methods by considering the distribution of parameters rather than a single fixed parameter value [23]. Since void swelling datasets are sparse, a Bayesian approach is a more suitable choice.

The first step of the Bayesian parameter estimation is to specify the prior distribution of the parameters, where the model parameters are represented by $\boldsymbol{\theta} = [\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3, \sigma^2, \boldsymbol{\Sigma}]$. Here, we use informative prior distributions [28] inspired by domain knowledge to overcome the imbalanced data challenge mentioned in Section 2.1.2. Ideally, given an incomplete unit in the transient regime, the proposed model should estimate the changepoint β_{CPi} to be located somewhere at the right of the measurements (high dpa values), and the opposite for incomplete units in the steady state regime.

The key insight in identifying the regime lies in the swelling rate. The definition of the regimes in Section 2.1.2 hints that the steady state swelling rate is in general significantly higher than the transient swelling rate. Hence, priors on $\boldsymbol{\gamma}_2$ should have a larger mean than priors on $\boldsymbol{\gamma}_1$ such that $\beta_{2i} > \beta_{1i}$. Furthermore, the prior on $\boldsymbol{\gamma}_2$ is set to have a larger variance term than the prior on $\boldsymbol{\gamma}_1$ to encourage β_{2i} capture more drastic swelling trends in the steady state regime. Accordingly, we set the priors $\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1 \sim N(0,50)$, and $\boldsymbol{\gamma}_2 \sim N(20,75)$. Finally, the prior on $\boldsymbol{\gamma}_3$ should guide α_{CPi} to stay in the range [-4,4] to prevent the inverse logit function from saturating to 0 or 1. Section 2.5.3 later investigates the proposed model's sensitivity to different prior distributions.

We impose uninformative prior distributions for the remaining parameters. For the unknown level 2 covariance matrix $\boldsymbol{\Sigma}$, the popular Lewandowski-Kurowicka-Joe (LKJ) prior [29] with parameter $\eta = 1$ (i.e., $LKJ(\eta)$) is imposed. The LKJ prior is a widely used prior distribution for correlation and covariance matrices, with the shape parameter η controlling the amount of correlation among the level 2 random errors [30]. Here, the LKJ prior essentially acts as a uniform prior over the correlation matrix. For the standard deviation of the level 1 random errors, σ, a half Student's t distribution prior with 3 degrees of freedom is used.

The second step in Bayesian parameter estimation is identifying the likelihood function. The likelihood function for the level 2 equation, $p(\boldsymbol{\beta}_i | \boldsymbol{Z}_i, \boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3, \boldsymbol{\Sigma})$, follows a multivariate normal distribution with mean $\boldsymbol{\Gamma}_i = [\boldsymbol{\gamma}_0^T \boldsymbol{Z}_i, \boldsymbol{\gamma}_1^T \boldsymbol{Z}_i, \boldsymbol{\gamma}_2^T \boldsymbol{Z}_i, Inv\ logit(\boldsymbol{\gamma}_3^T \boldsymbol{Z}_i)]^T \in \mathbb{R}^{4\times 1}$ and covariance matrix $\boldsymbol{\Sigma}$. For the level 1 equation, the likelihood function for a single observation Y_{ij}, $p(Y_{ij} | x_{ij}, \boldsymbol{\beta}_i, \sigma^2)$, also follows a normal distribution with mean μ_{ij} and variance σ^2, where $\mu_{ij} = \beta_{0i} + \beta_{1i}(x_{ij} - \beta_{CPi})\mathbb{I}\{x_{ij} < \beta_{CPi}\} + \beta_{2i}(x_{ij} - \beta_{CPi})\mathbb{I}\{x_{ij} \geq \beta_{CPi}\}$.

The third step is deriving the posterior distribution using Bayes' rule. The posterior distribution of the parameters is shown in equation (2.6), where the conditioning on $\boldsymbol{Z}_i$ and $\boldsymbol{x}_i$ is omitted for brevity.

$$p(\boldsymbol{\theta}|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = \int p(\boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\beta}_i)p(\boldsymbol{\beta}_i|\boldsymbol{\theta})d\boldsymbol{\beta}_i p(\boldsymbol{\theta}), \tag{2.6}$$

$$\text{Stage 1: } p(\boldsymbol{Y}|\boldsymbol{\beta}_i, \boldsymbol{\theta}) \propto \prod_{i=1}^{I}\prod_{j=1}^{n_i} \frac{1}{\sqrt{\sigma^2}} \exp\left\{-\frac{\left(Y_{ij} - \mu_{ij}\right)^2}{2\sigma^2}\right\},$$

$$\text{Stage 2: } p(\boldsymbol{\beta}_i|\boldsymbol{\theta}) \propto \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}_i - \boldsymbol{\Gamma}_i)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\beta}_i - \boldsymbol{\Gamma}_i)\right\},$$

Here, the $p(\boldsymbol{\theta})$ denotes the joint prior distribution of the model parameters $\boldsymbol{\theta}$. The posterior distribution in equation (2.6) is computationally intractable, and thus we use the No-U-Turn-Sampler (NUTS) [31], an extension of the conventional MCMC. Compared to traditional MCMC, NUTS uses gradient information to guide the algorithm and generates higher-quality samples with less autocorrelation much more quickly. The NUTS algorithm is implemented in the software Stan [30].

2.4.4 Prediction

Recall that our goal is to predict the unobserved swelling measurement Y_i^* of unit i with covariates $\boldsymbol{Z}_i$ at the new irradiation dose level x_i^*, i.e., to compute the posterior predictive distribution $p(Y_i^*|\boldsymbol{Y}, \boldsymbol{Z}, x_i^*)$. Note that $\boldsymbol{Y}_i = \left[Y_{i,1}; \ldots; Y_{i,n_i}\right] \in \mathbb{R}^{n_i \times 1}$ is the vector of historical measurements for unit i, where n_i is the total number of collected measurements for unit i, $\boldsymbol{Y} = [\boldsymbol{Y}_1; \ldots; \boldsymbol{Y}_I] \in \mathbb{R}^{(\Sigma n_i) \times 1}$ is the vector of historical measurements from all units, and $\boldsymbol{Z} = [\boldsymbol{Z}_1, ., ,, \boldsymbol{Z}_I] \in \mathbb{R}^{(P+1) \times I}$. Then, $p(Y_i^*|\boldsymbol{Y})$ can be expanded into the following form:

$$p(Y_i^*|\boldsymbol{Y}) = \iint p(Y_i^*|\boldsymbol{\beta}_i, \boldsymbol{\theta}) p(\boldsymbol{\beta}_i|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{Y}) d\boldsymbol{\beta}_i d\boldsymbol{\theta}, \tag{2.7}$$

where the conditioning over $\boldsymbol{Z}, \boldsymbol{Z_i}$ and $\boldsymbol{x_i^*}$ is omitted for brevity. The posterior predictive distribution is approximated numerically by using the Monte Carlo Integration:

$$p(Y_i^*|\boldsymbol{Y}) \approx \frac{1}{W} \sum_{w=1}^{W} p\left(Y_i^* \middle| \boldsymbol{\beta}_{w,i}, \boldsymbol{\theta}_w\right), \boldsymbol{\theta}_w \sim p(\boldsymbol{\theta}|\boldsymbol{Y}), \boldsymbol{\beta}_{w,i} \sim p(\boldsymbol{\beta}_i|\boldsymbol{\theta}_w), \tag{2.8}$$

in which $\boldsymbol{\theta}_w = [\boldsymbol{\gamma}_{w,0}, \boldsymbol{\gamma}_{w,1}, \boldsymbol{\gamma}_{w,2}, \boldsymbol{\gamma}_{w,3}, \sigma_w^2, \boldsymbol{\Sigma}_w]$ is the wth Monte Carlo sample drawn from the posterior distribution in Section 2.4.3 and W represents the total number of posterior samples, i.e., $w \in \{1, \dots, W\}$. Furthermore, $\boldsymbol{\beta}_{w,i} = \left[\beta_{w,0i}, \beta_{w,1i}, \beta_{w,2i}, \beta_{w,CPi}\right]^T$ represents the $\boldsymbol{\beta}_i$ values obtained based on equation (2.4) using $\boldsymbol{\theta}_w$. This way, $Y_i^*|\boldsymbol{\beta}_{w.i}, \boldsymbol{\theta}_w$ can be easily obtained as it follows a normal distribution with mean $\beta_{w,0i} + \beta_{w,1i}\left(x_i^* - \beta_{w,CPi}\right)\mathbb{I}\left\{x_i^* < \beta_{w,CPi}\right\} + \beta_{w,2i}\left(x_i^* - \beta_{w,CPi}\right)\mathbb{I}\left\{x_i^* \geq \beta_{w,CPi}\right\}$ and variance σ_w^2.

One special case is the cold start case where the new unit i' of interest has not collected any swelling measurements, i.e., $\boldsymbol{Y}_{i'} = \emptyset$ and $\boldsymbol{Z}_{i'} \notin \{\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_I\}$. Here, the distribution of interest is $p\left(Y_{i'}^* \middle| \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{Z}_{i'}, x_{i'}^*\right)$, in which $Y_{i'}^*$ is an unobserved swelling measurement from unit i' at the irradiation dose value of $x_{i'}^*$. The cold start posterior predictive distribution $p\left(Y_{i'}^* \middle| \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{Z}_{i'}, x_{i'}^*\right)$ is numerically approximated similarly as in equation (2.8), where the posterior draws of the parameters are made from *I* historical units:

$$p\left(Y_{i'}^* \middle| \boldsymbol{Y}\right) \approx \frac{1}{W} \sum_{w=1}^{W} p\left(Y_{i'}^* \middle| \boldsymbol{\beta}_{w,i'}, \boldsymbol{\theta}_w\right), \boldsymbol{\theta}_w \sim p(\boldsymbol{\theta}|\boldsymbol{Y}), \qquad \boldsymbol{\beta}_{w,i'} \sim p(\boldsymbol{\beta}_{i'}|\boldsymbol{\theta}_w). \tag{2.9}$$

Similar to equation (2.7) and equation (2.8), the conditioning over $\boldsymbol{Z}, \boldsymbol{Z}_{i'}$ and $x_{i'}^*$ are omitted for brevity. In equation (2.9), $\boldsymbol{\beta}_{w,i'}$ is calculated by using the posterior estimates of $\boldsymbol{\theta}_w$ and the covariates of the cold start unit $\boldsymbol{Z}_{i'}$ in equation (2.4). For instance, $\beta_{0i'} = \boldsymbol{\gamma}_{w,0}^T \boldsymbol{Z}_{i'} + u_{0i'}$ in which

$\boldsymbol{\gamma}_{w,0}$ is obtained using $\boldsymbol{\theta}_w$ and $u_{0i'}$ is randomly chosen from the pool of $\{u_{01}, \ldots, u_{0I}\}$ for each sample.

2.5 Numerical Studies

In this section, we evaluate the proposed model on a real-life void swelling dataset. In Section 2.5.1, we introduce five benchmark methods. Section 2.5.2 then discusses the model validation approaches used in this study. Then, Sections 2.5.3 and 2.5.4 contain the results of our numerical analysis.

2.5.1 Benchmark Methods

We briefly discuss the five benchmark methods that will be used in the model evaluations. First, we consider a linear regression model with an L2 regularization term. For a fair comparison, the coefficients of the linear regression are constrained to be positive in order to enforce the monotonic relationship of void swelling. Here, the predictors are the irradiation dose and covariates, while the response variable is the void swelling %.

For the second approach, we consider a set of ensemble methods. The main idea behind ensemble learning is to combine the prediction of several base estimators to obtain better performance than using a single estimator [32]. Generally, ensemble methods are divided into averaging methods and boosting methods. Here, we consider one model from each category as a benchmark: a Random Forest (RF) [33] for averaging methods and a Gradient Boosted Trees (GBT) [34] for boosting methods.

The next benchmark is the ANN model [35]. ANN has received much attention in the past years due to its strong predictive performance and flexibility. ANN is a network of neurons (i.e., nodes) with input, hidden, and output layers. Contrary to models that can only capture linear relationships,

ANN can capture complex and nonlinear relationships by employing nonlinear activation functions. Furthermore, we constrain the weights and biases of the ANN to have a monotonic relationship between the irradiation dose and void swelling %.

The last benchmark is the Multioutput Gaussian Process (MGP) regression [36]. A GP is a collection of random variables where any finite number of which has a multivariate Gaussian distribution. Although GPs are very flexible and effective at modeling arbitrary functions, they exhibit poor performance in extrapolation tasks. To address this limitation, we adopt a MGP with a *separable* covariance structure proposed by Bonilla, Chai, and Williams [36], which can transfer information across different units to improve extrapolation performance.

For all benchmark methods, the input is the concatenated vector of the irradiation dose value and covariates. In other words, the input for the jth measurement of unit i is the vector $\left[x_{ij}, Z_{1i}, \dots, Z_{Pi}\right]^T \in \mathbb{R}^{(P+1)\times 1}$ and the output is Y_{ij}. The obtained predictions $\hat{Y}_{ij}$ are then evaluated on three metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE). Equation (2.10) lists the exact formulas for each metric in detail:

$$MAE = \frac{\sum_{i=1}^{I_{test}} \sum_{j=1}^{U_i} \left|Y_{ij} - \hat{Y}_{ij}\right|}{\sum_{i=1}^{I_{test}} U_i}, MSE = \frac{\sum_{i=1}^{I_{test}} \sum_{j=1}^{U_i} \left(Y_{ij} - \hat{Y}_{ij}\right)^2}{\sum_{i=1}^{I_{test}} U_i}$$
$$MAPE = \frac{\sum_{i=1}^{I_{test}} \sum_{j=1}^{U_i} \left|(Y_{ij} - \hat{Y}_{ij})/Y_{ij}\right|}{\sum_{i=1}^{I_{test}} U_i} \tag{2.10}$$

where U_i is the number of unobserved swelling measurements for unit i, and I_{test} is the total number of test units. All benchmark methods are implemented in Python, while the proposed Bayesian hierarchical piecewise linear regression model is implemented in the R package *brms* [37] with a Stan backend [30]. The detailed parameter settings used in the evaluations can be found in the supplementary materials.

2.5.2 Model Validation Methods

In this subsection, we discuss the methods used for checking the proposed Bayesian model. First, we check the validity of the Bayesian approach by comparing the prediction results with maximum likelihood estimation. Then, we check the adequacy of the prior distribution of the mean of the changepoint parameter $\boldsymbol{\gamma}_3$, which is essential in correctly identifying the regime information. Finally, we examine the Bayesian model fitting results by checking the posterior predictive distribution and computing the LOO-PIT (leave one out probability integral transform) values.

As mentioned in Section 2.4.3, the prior distribution on the changepoint parameter $\boldsymbol{\gamma}_3$ affects the estimation of the changepoint between the transient and the steady state regimes. To assess the influence of the prior distribution, we first determine the true regime information by examining the relevant literature. For example, it is known that the 316 stainless steels treated with Carbon and Nitrogen solutions have a steady state that begins after 0.02 dpa with normalization [38]. Similarly, the dpa value at which voids are first observed can be used as the changepoint. After obtaining the true regime information of the test units, we assess the regime prediction accuracy in two ways. For the incomplete units, the predictions are evaluated by the percentage of correctly predicted regions. For the complete units, the predictions are evaluated by MAE between $\hat{\beta}_{CPi}$ and β_{CPi}.

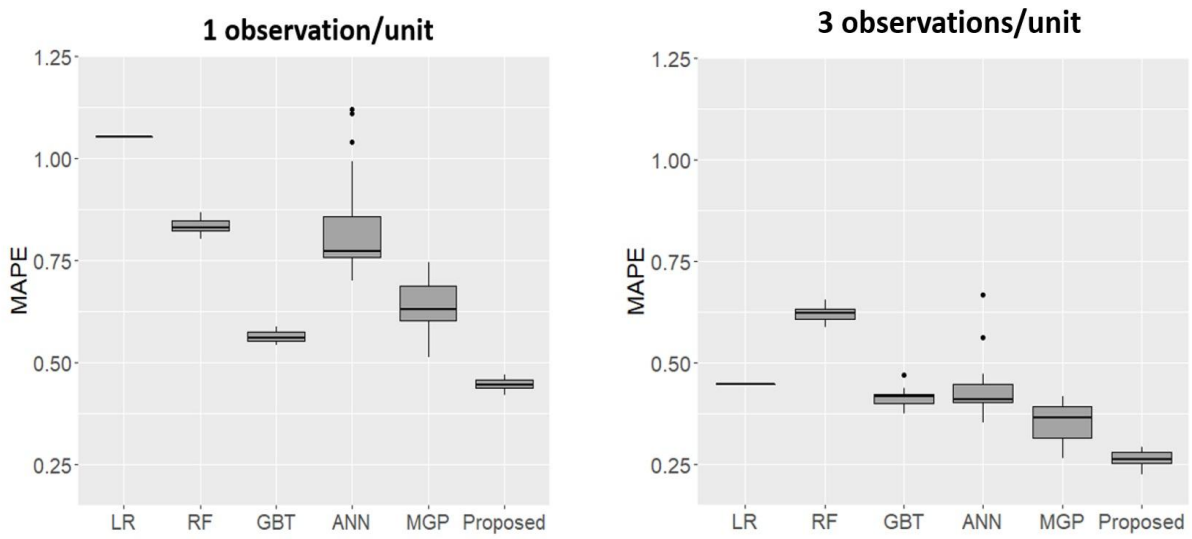
Finally, we check the adequacy of the model fit by examining the posterior predictive distribution. Posterior predictive checking essentially compares the distribution of the true void swelling values Y_{ij} and the simulated void swelling values Y_{ij}^{rep} from the HMC algorithm. The LOO-PIT method calculates the probability distribution for each marginal prediction separately, and then compares these separate distributions to the existing data distribution to check model calibration or find outliers.

2.5.3 Evaluation Results: Scenario 1 (Partially Observed Units)

In this subsection, we compare the performance of the proposed model to the benchmark methods. Specifically, we assume that the training units have access to all measurements, while the test units have partial access to the first few measurements. Based on the available measurements, our task is to predict the unobserved (hidden) void swelling % of the test units. To begin, the units are first split into training and test units. In particular, the 12 units that have clear observable swelling trends with more than 3 observations are divided into train/test sets based on 4-fold cross-validation. The remaining 279 units with 3 or fewer observations are always regarded as training units. As a result, each evaluation iteration has 279+9=288 training units and 3 testing units. For all evaluations, the NUTS ran with 4 chains and 6000 iterations, in which the first 3000 iterations were used as the warm-up stage. Also, the evaluations are repeated 50 times. Note that the accuracy of the NUTS sampler and its integration process are evaluated via posterior predictive checking and LOO-PIT values. The detailed results are provided in the supplementary materials.

Table 2.2 Evaluation results for scenario 1, measured by Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE) (Boldface: Lowest error, Paranthesis: Error standard deviation).

	1 observation/unit			**3 observations/unit**		
Model	**MAE**	**MSE**	**MAPE**	**MAE**	**MSE**	**MAPE**
Proposed Model	**1.4362** (0.1390)	11.476 (0.4092)	**0.4418** (0.0148)	**1.3948** (0.0549)	9.1789 (0.4938)	**0.2526** (0.0214)
Linear Regression (LR)	1.9212	13.781	1.0529	1.8614	15.593	0.4474
Random Forest (RF)	2.2086 (0.0644)	10.379 (0.4623)	0.8326 (0.0191)	2.4851 (0.0663)	11.811 (0.6560)	0.6204 (0.0181)
Gradient Boosted Trees (GBT)	2.0691 (0.0825)	**8.5278** (0.6611)	0.5646 (0.0140)	1.9421 (0.0707)	**7.3637** (0.3912)	0.4128 (0.0221)
Artificial Neural Network (ANN)	1.6988 (0.1189)	12.242 (0.9687)	0.8360 (0.1256)	1.7593 (0.3072)	14.648 (1.4157)	0.4330 (0.0684)
Multivariate Gaussian Process (MGP)	2.9163 (0.2141)	22.665 (3.3855)	0.6407 (0.0612)	2.5847 (0.4289)	19.983 (5.1449)	0.3524 (0.0489)



(a) (b)

Figure 2.7 Box plot of MAPE results for scenario 1

(a) 1 observation/unit (b) 3 observations/unit

The results of the "partially observed" scenario are organized in Table 2.2, in which the term "observation/unit" denotes the number of initially observed measurements from the test units. For instance, "1 observation/unit" indicates that the first measurement of the test units is observed and then we predict the remaining future measurements of the test unit. For visual clarity, the lowest errors (i.e., best-performing model) in each category are boldfaced. The standard deviations of the errors are shown in parenthesis. In addition, the error distribution in terms of MAPE for each method is shown in Figure 2.7.

In addition to the error metrics, the predicted swelling trends must be monotonically nondecreasing (i.e., there is no decrease in swelling except for small measurement errors) with a clear identification of the changepoint when applicable. Figure 2.8 and Figure 2.9 show the predicted swelling curves for two sample units with normalized dpa values. The first unit (id = 2) shows an incomplete unit with only one identifiable regime, while the second unit (id = 7) shows a complete unit with two identifiable regimes.

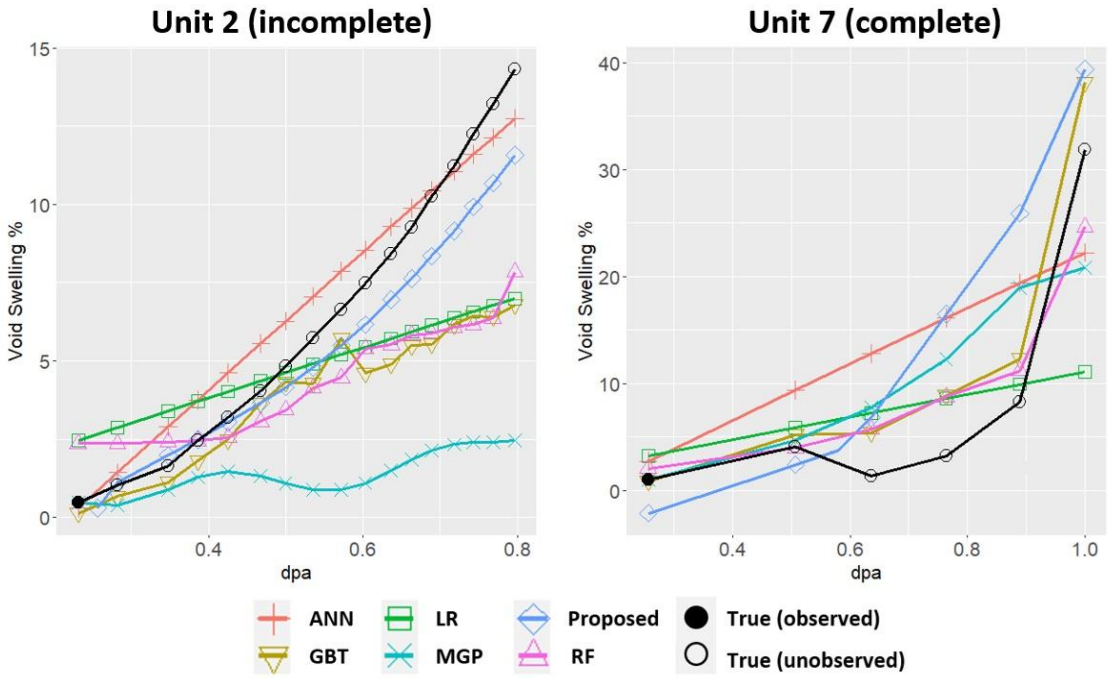


Figure 2.8 Benchmark method plots from unit 2 and unit 7 (1 observation/unit)

The results from Table 2.2 highlight the superior performance of the proposed model over existing benchmark methods in terms of MAE and MAPE. For other benchmark methods, the ANN and LR methods' accuracy suffers due to their positive constraints, while the RF, GBT, and MGP methods frequently result in erroneous predictions. For instance, the GBT, MGP, and RF predictions for Figure 2.8 in unit 2 all return locally decreasing swelling trends. On the contrary, the proposed method returns predictions that are accurate and coherent with the properties of void swelling (i.e., nondecreasing and clearly identifies two trends when applicable). The proposed model performs slightly less than the GBT in terms of MSE. However, it is important to highlight that MSE is a less stable metric than MAE or MAPE as it tends to exaggerate the errors made by outliers due to the squared term. In addition, the GBT had higher percentage errors in earlier transient regimes, resulting in higher MAPE values. Also, we confirm from the swelling trajectories of unit 2 and unit 7 that the model predictions improve as more measurements are available. For instance, unit 7's predictions in Figure 2.8 with 1 observation/unit identify the

changepoint to be located around 0.5 dpa. However, in Figure 2.9 with 3 observations/unit, the model predicts that the changepoint is around 0.8 dpa, which is coherent with the true changepoint value.

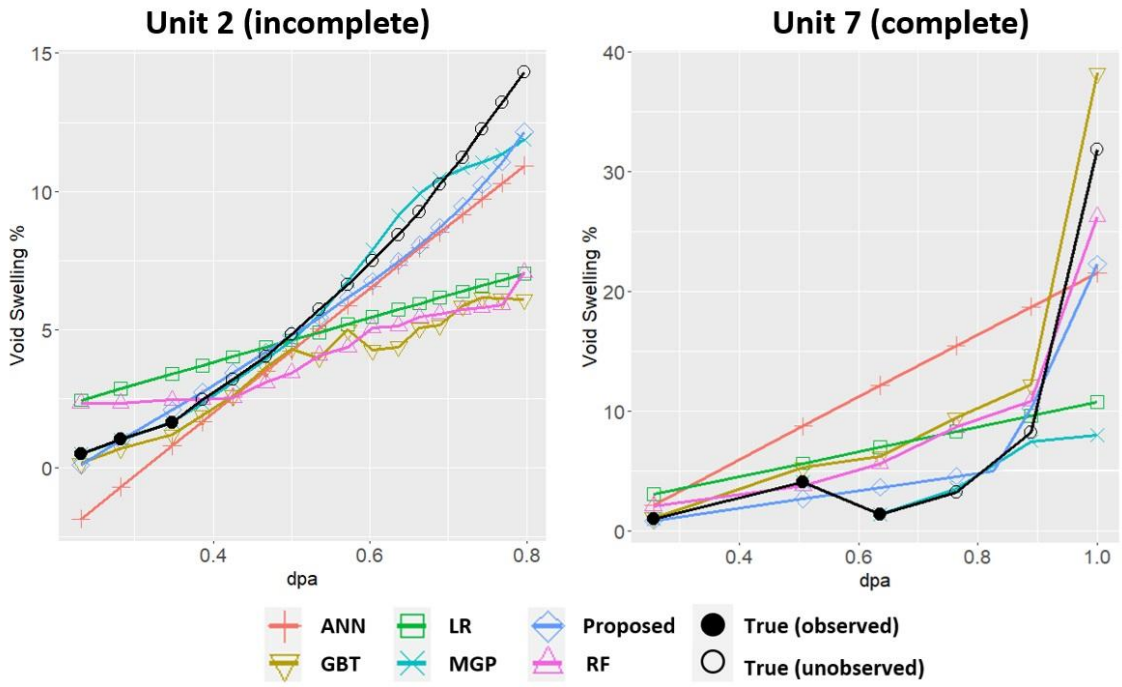


Figure 2.9 Benchmark method plots from unit 2 and unit 7 (3 observation/unit)

We first compare the Bayesian predictions from the MLE predictions. Figure 2.10 illustrates the predicted swelling curves with the posterior predictive distribution of a randomly selected test unit assuming different levels of data availability. Note that the posterior samples outside the 2.5th and 97.5th quantiles are neglected to remove the effect of extreme outliers. The figure shows that the predictions from the ML approach suffer from high bias. On the contrary, the predictions from the Bayesian approach accurately capture the uncertainties in its predictions with the prediction intervals covering the true swelling values, and the mean predictions are much closer to the true values. The prediction errors of each parameter estimation method are shown in Table 2.3, representing the mean and standard deviation values derived from 50 repetitions. The results

demonstrate the superior performance of the Bayesian approach over the maximum likelihood approach. The lowest errors are boldfaced for visual clarity.

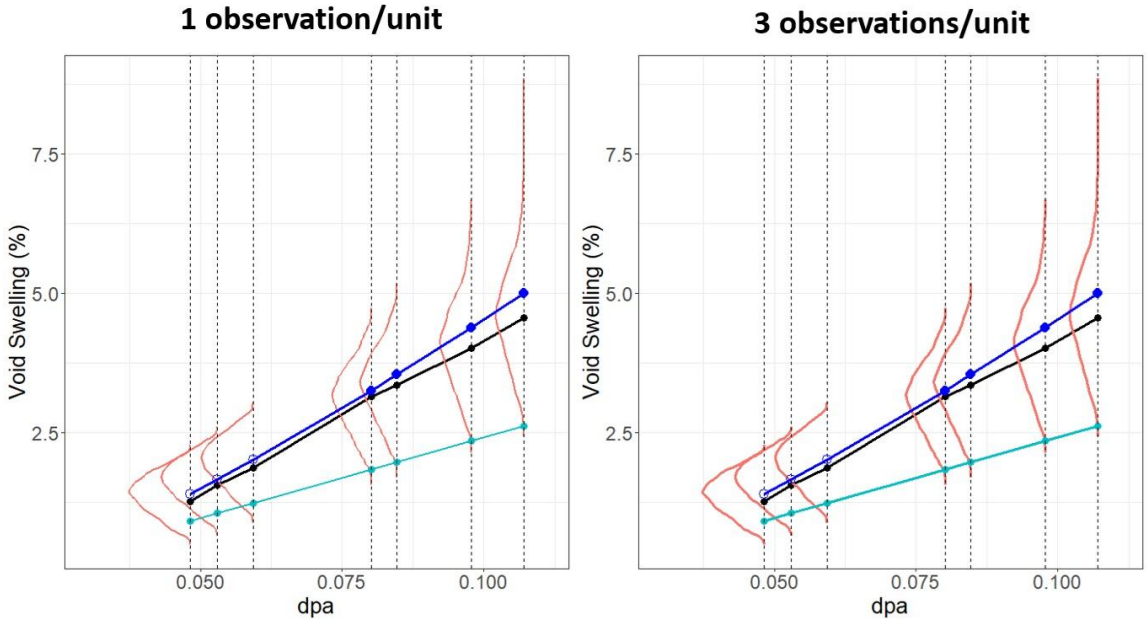


Figure 2.10 Proposed model's predictions vs. MLE predictions for a sample unit (id = 3) (Black: True, Blue: Proposed Model, Red: Prediction intervals (Proposed Model), Jade: ML)

Table 2.3 Evaluation results for Bayesian parameter estimation vs. maximum likelihood, measured by MAE, MSE, and MAPE. (Boldface: Lowest error, Paranthesis: Error standard deviation).

	1 observation/unit			**3 observations/unit**		
Model	**MAE**	**MSE**	**MAPE**	**MAE**	**MSE**	**MAPE**
Proposed Model	**1.4362** (0.1390)	**11.476** (0.4092)	**0.4418** (0.0148)	**1.3948** (0.0549)	**9.1789** (0.4938)	**0.2526** (0.0214)
Maximum Likelihood (ML)	2.3068	15.472	0.6947	2.2297	13.503	0.5644

Finally, we examine how the proposed model identifies the regimes via informative prior distributions. Out of the 12 test units with more than 3 observations, 10 units are incomplete with only one regime information, and 2 units are complete with both regimes. All 10 incomplete units are determined to be in the steady state regime.

Table 2.4 Accuracy of the proposed method in determining regimes with (un-)informative priors on $\boldsymbol{\gamma}_3$

Metric\Prior	**Uniform(-4,4) (Uninformative)**	**Normal(0,2) (Informative)**
% Correct (incomplete)	34.20% (4.9857)	80.80% (8.533)
MAE (complete)	0.2948 (0.0218)	0.1286 (0.0312)

The regime prediction results are shown in Table 2.4, where the evaluations are repeated 50 times, and the standard deviations are reported in parenthesis. Note that the Uniform(-4,4) prior represents an uninformative prior, while Normal(0,2) is a more informative prior with tailored variance parameters to control the values of $\hat{\beta}_{CPi}$. Results of Table 2.4 indicate that the proposed model is sensitive to the choice of the prior distribution on $\boldsymbol{\gamma}_3$, highlighting the importance of incorporating domain knowledge by choosing informative priors.

2.5.4 Evaluation Results: Scenario 2 (Cold Start Units)

The second scenario is the "cold start" unit that was introduced at the end of Section 2.4.4. Recall that the cold start unit only has access to the covariate information. Since there are no past measurements to estimate the void swelling trajectory, it is much more difficult for traditional data-driven methods to accurately estimate the trend of a cold start unit.

Analyzing cold start units holds great potential for practitioners. For example, suppose that a researcher is interested in investigating the effect of irradiation temperature on the swelling of a new type of austenitic steel. Instead of manually conducting expensive experiments, researchers can simply plug in the cold start unit's covariates into the proposed model and examine the predicted void swelling trends.

Since this demonstration is a proof-of-concept, the predictions are evaluated on whether the predicted swelling curves follow the same trend as the true swelling curves. To combat the limited data availability of the cold start setting, numerical experiments are conducted under a leave-one-

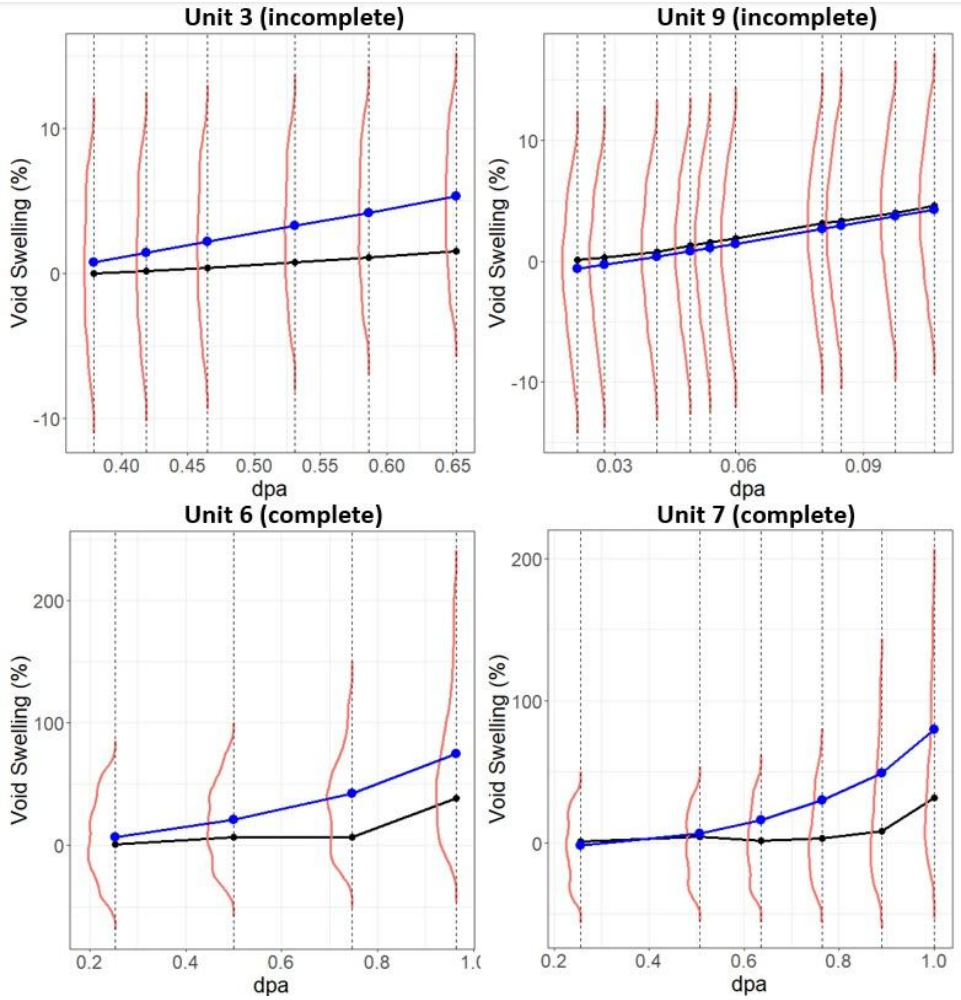


Figure 2.11 Scenario 2 predictions (cold start)

(Black: True, Blue: Proposed Model, Red: Prediction intervals (Proposed Model))

out cross-validation setting. In other words, in each cross-validation iteration, we have one test unit (with no available measurements) and the remaining 11+279 = 290 units as training units (with all measurements available). To clearly observe the swelling trends, we plot the predictions for 4 units (2 incomplete and complete units each) alongside the posterior predictive distribution in Figure 2.11. From the figure, although the range of the posterior predictive distribution is wider than that of scenario 1, most of the probability mass is located near the true swelling values. The

results show that the covariate information alone can provide valuable guidance for the proposed model to accurately estimate the general swelling trends.

2.6 Discussion & Conclusion

In this case study, we have considered a novel data-driven Bayesian piecewise hierarchical linear regression to directly model and predict the degradation status of materials subject to void swelling. The proposed method has the following contributions. First, it incorporates domain knowledge of void swelling by imposing specific shape constraints. Second, the joint effect of multiple covariates is naturally represented through the hierarchical structure of the model. Finally, the proposed model overcomes the limited availability of the swelling dataset by leveraging the advantages of a Bayesian approach. In particular, the uncertainty quantifications of the predicted swelling values to assess the reliability of predictions.

Numerical studies on a real-life void swelling dataset showed that the proposed model outperforms traditional data-driven models such as LR, ensemble methods, MGP, and ANN in predicting the swelling values. In addition, the estimated 95% credible intervals included the true swelling values for all units. Even for cold start units, the proposed model still managed to provide reasonable estimates of the swelling trends. Overall, the proposed method has demonstrated the effectiveness of a data-driven method tailored for void swelling and the potential to be used as a reference for practitioners. Furthermore, the predicted degradation status can be used to construct early warning indicator systems that can greatly aid NPP maintenance and prevent unexpected failures and catastrophic accidents.

The proposed Bayesian hierarchical model is not just limited to void swelling and industrial applications and can also be used in a wide range of systems with nested/hierarchical data structures. Here, although we choose the level 1 regression shape to be a piecewise linear trend,

the model can be easily modified to accommodate various trends. In addition, prior knowledge of the engineering system can be incorporated into the model by imposing constraints on the regression parameters. In summary, we hope that this case study will catalyze future research that uses Bayesian hierarchical models to perform predictive maintenance and degradation modeling in both traditional and nontraditional applications.

Future studies will focus on the following topics. One potential topic is optimal covariate design to extend the swelling period of a unit to its maximum extent. This is also known as informed alloy design, in which we can employ optimization techniques to identify optimal covariates that elongate the lifetime of a unit as much as possible. Extending the lifetime of a unit can result in better experimental design and aging management in nuclear facilities. Second, we can consider active learning techniques to adaptively obtain experimental measurements. For example, using the entropy criterion, we can collect subsequent swelling measurements at the irradiation dose level with the highest entropy (i.e., largest predictive uncertainty). As we make informed sampling decisions guided by active learning, we can achieve more accurate estimations of swelling processes with a given number of measurements.

2.7 Supplementary Materials

2.7.1 Parameter Settings for Benchmark Methods

This section lists the parameter settings for the benchmark methods used in the evaluations in Section 2.5. The model parameters in Table 2.5 were optimized via K-fold cross validation with K = 4 except for the monotonic ANN and MGP.

Table 2.5 Optimized model hyperparameters.

Model	Parameters
Sklearn.linear_model.Ridge (Monotonic Linear Regression)	Alpha = 0.1 Positive = True
Monotonic ANN	Activation: ReLU Early Stopping: disabled Optimizer: Adam Epochs = 100 Learning Rate: 0.001 Batch Size = 32 Layer Size = [40,40]
Sklearn.ensemble.RandomForestRegressor (Random Forest)	N_estimators = 80 Max_depth = 7
Sklearn.ensemble.GradientBoostingRegressor (Gradient Boosted Trees)	N_estimators = 50 Max_depth = 8
Multioutput Gaussian Process	Covariance = Rational Quadratic

2.7.2 Model Adequacy Checking

The adequacy of the proposed model is evaluated using the posterior predictive checking and LOO-PIT (leave one out probability integral transform). Posterior predictive checking overlays the simulated densities of Y_{ij}^{rep} with the true density of Y_{ij}. Here, Y_{ij} represents the jth swelling measurement of unit i. Figure 2.12 and Figure 2.13 show the posterior predictive checking plots from the 4-fold cross-validation. The results demonstrate that the simulated densities overlap with the true densities, suggesting that the proposed model has a good fit.

The second method for model adequacy checking is the LOO-PIT values. The LOO-PIT method calculates the probability distribution for each marginal prediction separately, and then compares these separate distributions to the existing data distribution to check model calibration or detect outliers. A model with good fit typically shows no major deviations with asymptotically symmetric trends in the LOO-PIT plots. Again, we observe from Figure 2.14 and Figure 2.15 that there are no significant outliers from the calculated LOO-PIT plots, further showing that the model has a good fit.

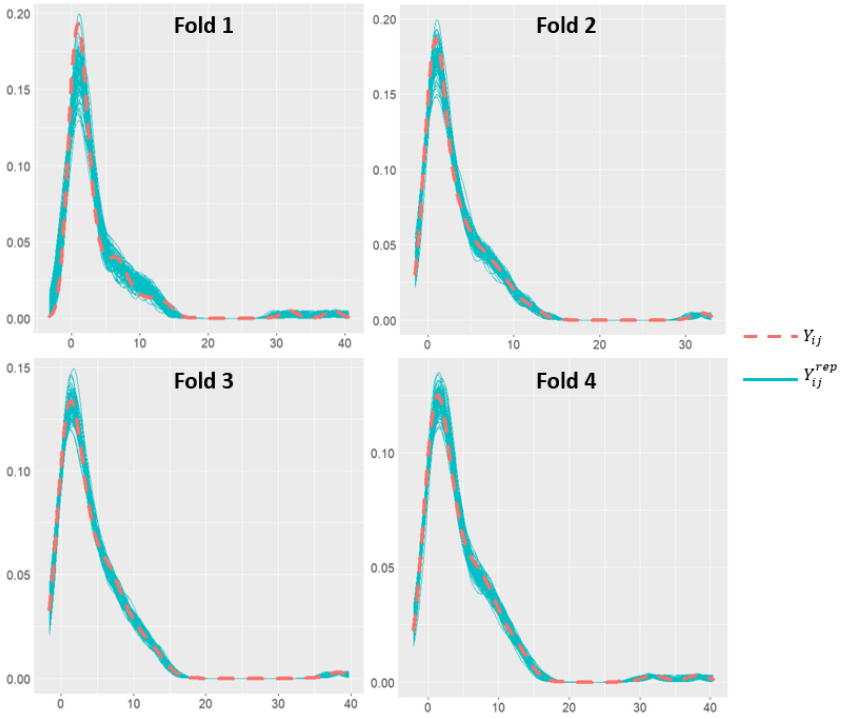


Figure 2.12 Posterior predictive checking (Red dashed: Y_{ij}, Jade solid: Y_{ij}^{rep})

1 observations/unit

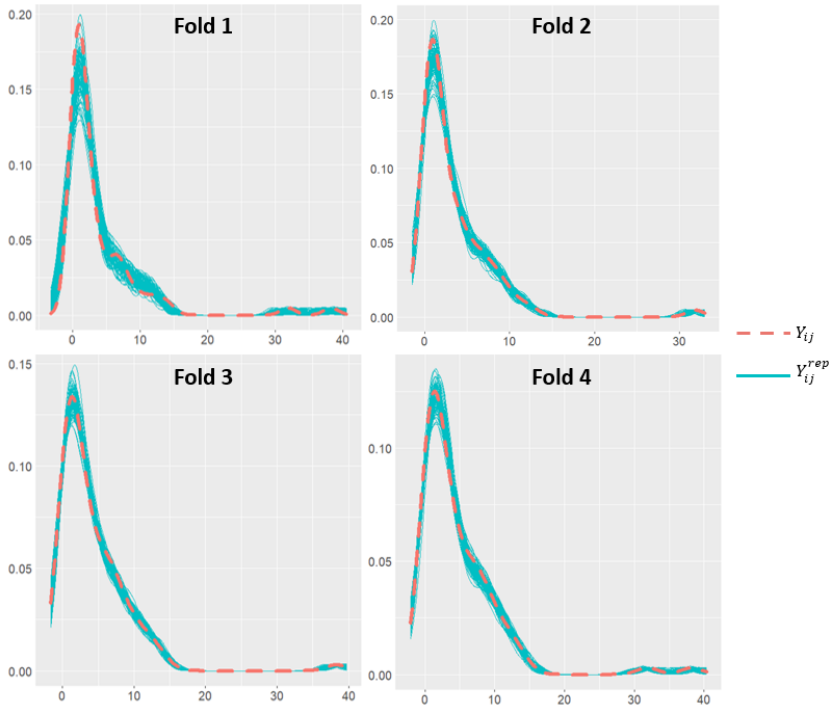


Figure 2.13 Posterior predictive checking (Red dashed: Y_{ij}, Jade solid: Y_{ij}^{rep})

3 observations/unit

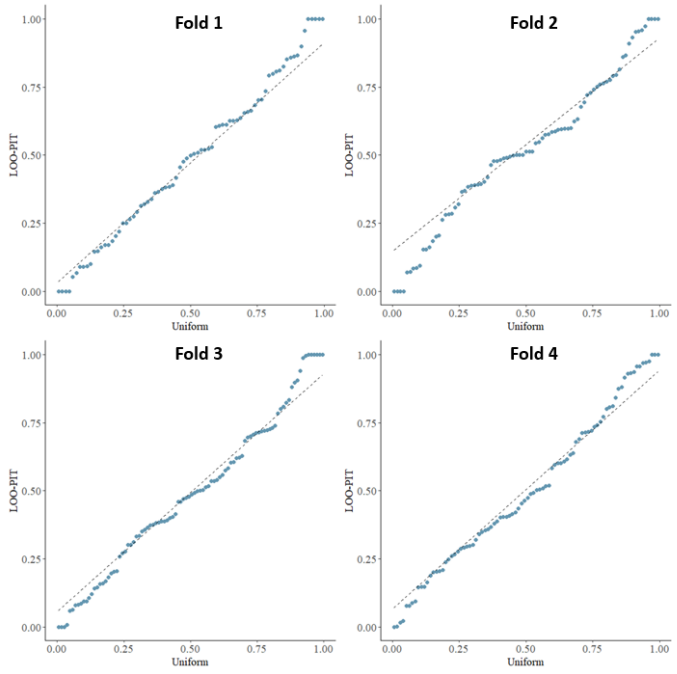


Figure 2.14 LOO-PIT curves, 1 observation/unit

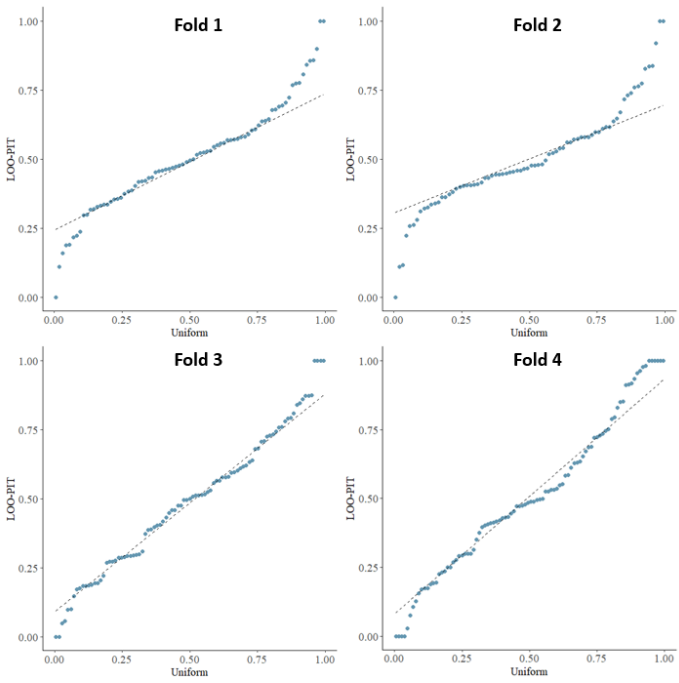


Figure 2.15 LOO-PIT curves, 3 observations/unit

2.7.3 Recommendations for Choosing the Prior Distributions

Here, we provide guidelines on how to choose the appropriate prior distributions for the model parameters. First, one must decide between noninformative and informative prior distributions. Generally, it is recommended to use informative priors when there is strong prior belief about the parameters. If there is no empirical studies or expert knowledge to draw insights from, then a noninformative prior distribution is recommended. Second, we select the probability distribution based on the nature of the parameters, such as bounded versus unbounded distributions, or positive constraints on the parameter values. Finally, after specifying the appropriate prior distributions, we then perform sensitivity analysis by trying different priors.

Chapter 3 An Integrated Uncertainty Quantification Model for Longitudinal and Time-to-event Data

3.1 Introduction

Recent advances in sensor technologies have led to the widespread use of multiple sensors to simultaneously monitor the various aspects of the condition of an engineering system and obtain an accurate diagnosis and prognosis of a system [7], which have significantly improved system profitability and reliability by preventing unexpected failures.

To accurately characterize the health of a system and predict the remaining useful life (RUL), existing approaches generally extract prognostic insights from either longitudinal sensor data or time-to-event/failure data. An example of such longitudinal data is multisensor signals (e.g., vibration, temperature, pressure) from manufacturing systems. Meanwhile, event data of the same system are typically available in the form of machine failure/maintenance logs. Each data type offers distinct perspectives on the health of a system. While longitudinal data contains information on signal evolution and temporal patterns [39], time-to-event data provides insights into censoring and the occurrence of failure events [40]. However, most current approaches do not take full advantage of the insights in both data types and instead opt to analyze one data type separately. Specifically, approaches based solely on longitudinal data typically define failure as an event when the degradation signal (or a function of degradation signals) reaches a fixed, pre-determined failure threshold (i.e., *soft failure*) [41]. Once a soft failure occurs, the system's performance is no longer considered to meet the required standards. A disadvantage of this assumption is that it can be difficult to define an exact failure threshold value in practice due to unit-to-unit variability and

multiple failure mechanisms. On the other hand, approaches based only on time-to-event data define failure in terms of risk (i.e., hazard) of failure (i.e., *hard failure*) [42].

To harvest the benefits of both longitudinal and time-to-event data, many researchers have explored joint models to simultaneously analyze both types of data. Joint models simultaneously incorporate both data types by first modeling the degradation signals via a mixed-effects model and then uses the fitted signals as time-varying covariates of the Cox PH model. Joint models do not require a pre-determined failure threshold as it naturally describes the failure probability via the hazard function of the Cox PH model. In addition, joint models provide unit-level modeling of the failure times and RUL while continuously tracking the evolution of the degradation signals.

For instance, Liao et al. [43], first used the linear Cox model with logistic regression to predict the lifespan of a bearing. Later, Zhou et al. [40] proposed a joint model with Bayesian updating to predict the RUL of automotive lead-acid batteries.

Unfortunately, a key drawback of existing joint models is their heavy reliance on restrictive parametric assumptions. The mixed-effects model requires one to predefine the functional form of the longitudinal data. As a result, it is susceptible to model misspecification errors and struggles to capture complex degradation trends. In addition, the linear Cox model is also limited by its strong parametric assumption. In particular, its linear-risk assumption restricts the model to only capture linear interactions between the log-hazard function and the covariates.

To relax the parametric assumptions, recent efforts have replaced the mixed-effects model with nonparametric methods. For instance, Yue and Kontar [44] proposed to model the longitudinal data by a multivariate Gaussian convolutional process (MGCP). While the flexibility of the MGCP has shown great potential in capturing the unit-to-unit variability in the signal trajectories similar to the proposed method, it suffers from drastically increased computational and storage costs.

Hence, it is less appealing for online RUL estimation especially based on large datasets. Zhou et al. [45] used a functional principal component analysis (FPCA)-based approach to model the signal trajectories. Although this nonparametric method is faster than the MGCP, it still ignores the degradation information embedded in the time-to-event data.

On the other hand, to overcome the linear-risk assumption in the linear Cox model, researchers have proposed other survival models with different structures. For instance, survival trees adopt a nonparametric, tree-based approach to model the interactions between the covariates and the log-risk function [46]. Other examples include accelerated failure time (AFT) models, which assume a linear relationship between the covariates and the log-transformed failure time [47]. Despite the relaxed modeling assumptions, both models directly use the longitudinal signals as time-varying covariates, which is known to result in biased and error-sensitive estimations.

Another approach to overcome the linear-risk assumption is to use a neural network (NN) to allow the modeling of nonlinear covariate interactions. Since NN can easily model arbitrary functions, NN-extended Cox models have received more attention in recent years and have outperformed traditional survival models in a variety of clinical applications, e.g., DeepSurv [48], PyCox [49], and SurvivalNet [50].

Despite their increased flexibility and predictive performance, directly using the predictions from NN-extended Cox models can result in detrimental errors for prognostics. First, a major limitation of these approaches is that they rely on a fixed, deterministic NN without uncertainty quantification to model the covariate interactions. This can be problematic for degradation applications, as limited data availability and the inherently stochastic nature of degradation processes make it impractical to only provide point RUL estimates with absolute certainty [10]. Moreover, deterministic NNs are prone to overfitting, especially when the amount of training data

is limited [51]. To ensure that the NN-extended Cox models provide accurate and reliable modeling of the degradation process, it is critical to quantify the uncertainties involved in the RUL predictions. Indeed, practitioners can leverage the RUL uncertainty quantifications to assess the quality of the predictions and detect abnormal model behavior. In addition, uncertainty quantifications can help stakeholders make better informed decisions [52]. For instance, a prognostic model with uncertainty quantification may predict that there is a 90% probability that the RUL lies between 800 and 1200 hours. If the cost of unexpected failure is high (e.g., nuclear power plant), we may choose to conduct maintenance activities early (e.g., around 800 hours). On the other hand, if the corrective maintenance cost is low while the preventive maintenance cost is high, we may plan maintenance activities later (e.g., around 1200 hours) to avoid unnecessary expenses. Hence, it is highly desirable to have RUL predictions with accurate uncertainty quantifications.

However, uncertainty quantification in joint models is challenging due to the complex model structure. Indeed, joint models contain two sub-models for each type of data, and propagating uncertainties across these sub-models with varying model parameters is no trivial task. Currently, existing methods only offer incomplete uncertainty quantifications by considering the uncertainties from only either the longitudinal sub-model [40], [44], [45], [53] or the time-to-event sub-model [54]. For instance, Wen et al. [53] recently proposed an advanced joint model (referred as “NN-Joint”) in which the longitudinal data is modeled via a mixed-effects model, and the time-to-event data is modeled by a NN-extended Cox model. Although the NN-Joint model achieved satisfactory results by relaxing the linear-risk assumption, its predictions neglect the uncertainties in the time-to-event sub-model. Thus, there is still a lingering demand for a more comprehensive framework that can deliver integrated uncertainty quantifications for both data types.

To fill this research gap, we present a flexible, integrated uncertainty quantification model (referred as "IUQ" hereafter) for the joint analysis of longitudinal and time-to-event data. The proposed IUQ model has two parts: a nonparametric FPCA-based model for the longitudinal data, and a Bayesian Neural Network-based Cox model (i.e., BNN-Cox) model for the time-to-event data. The major advantages of the proposed IUQ model are as follows. First, the proposed IUQ model provides well-quantified, integrated uncertainty estimates by integrating uncertainties across the two sub-models. To the best of our knowledge, this is the only model in the literature that systematically integrates the uncertainties involved in jointly modeling both longitudinal and time-to-event data. Second, the IUQ model allows more flexibility in modeling both types of data since FPCA and BNN do not impose strong parametric assumptions. As a result, the IUQ model can well characterize a variety of degradation signal trajectories and covariate interactions. Third, the proposed model allows online updating of the RUL distribution. Similar to existing joint models [3], [16] the proposed IUQ model can continuously update the RUL distribution and make

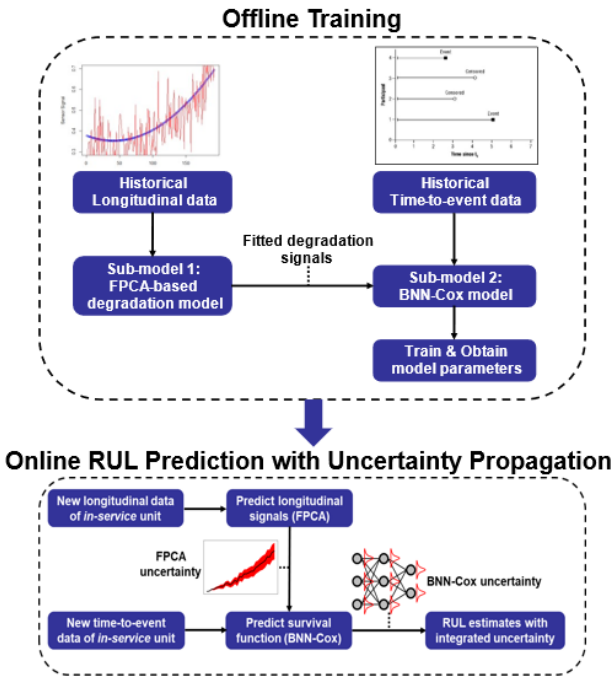


Figure 3.1 Overview of proposed joint modeling approach

highly individualized, real-time RUL predictions of an in-service unit based on its new observations. Finally, the proposed model provides reliable RUL estimates across varying levels of data availability.

Figure 3.1 illustrates the overall framework of the proposed joint modeling approach. In the offline stage, the FPCA-fitted degradation signals are fed into the BNN-Cox sub-model as time-varying covariates along with time-to-event data to estimate the model parameters. For the online stage, the newly collected measurements from the in-service unit are used to update the FPCA parameters via a Bayesian scheme. The calculated posterior FPCA parameters are then used to predict future degradation signals. In particular, the uncertainties in the FPCA parameters are integrated with the uncertainties in the BNN-Cox parameters to provide accurate survival and subsequent RUL estimates of the in-service unit in real time.

The remainder of the paper is organized as follows. In Section 3.2, we provide the details of the proposed joint modeling framework, offline parameter estimation, online updating procedures, and model prediction with uncertainty quantification. In Section 3.3, we conduct evaluations on both synthetic and real-life data. Finally, Section 3.4 summarizes the proposed method with its contributions and discusses future work.

3.2 Methodology

In this section, we will introduce the proposed IUQ model in detail. Sections 3.2.1 and 3.2.2 each describe the FPCA sub-model for longitudinal data and BNN-Cox sub-model for time-to-event data. Section 3.2.3 then elaborates on the offline training and parameter estimation. Finally, Section 3.2.4 discusses online RUL prediction with the uncertainty integration approach.

3.2.1 Sub-model 1: FPCA-based Degradation Modeling for Longitudinal Data

Here, we discuss the formulation of the FPCA-based degradation sub-model for longitudinal data. FPCA is one of the most popular dimensionality reduction methods for analyzing functional/longitudinal data. Specifically, FPCA assumes that the longitudinal data can be decomposed into a linear combination of orthonormal basis functions (i.e., eigenfunctions) and coefficients (i.e., FPC scores). Here, the eigenfunctions are chosen to explain the dominant modes of variation within the observed longitudinal data.

While several nonparametric approaches such as the Gaussian processes (GP) or splines can be employed to model longitudinal data, we choose to use FPCA due to its several practical benefits. First, performing FPCA is computationally less demanding than GP-based methods as it does not require the inversion of a large covariance matrix. Second, FPCA is effective at handling sparse and irregularly observed data [55], which is common in many degradation applications. Third, we can quantify the uncertainties in the longitudinal signals during online prediction by deriving the posterior distribution of the FPC scores.

Suppose that longitudinal data are collected over a compact time domain $\mathcal{T} \in [0, T_{max}]$, where T_{max} is the maximum possible event or failure time. The longitudinal data are assumed to be generated from a square-integrable stochastic process $Y(t)$ with mean function $\mu(t)$ and covariance function $\Sigma(t, t') = Cov\big(Y(t), Y(t')\big), (t \neq t')$. Mercer's theorem implies that the covariance $\Sigma(t, t')$ can be expanded into an infinite sum of eigenfunctions $\phi_k(t)$ and eigenvalues λ_k for $k = 1,2, \ldots$ under the linear Hilbert-Schmidt operator $G: L^2(\mathcal{T}) \to L^2(\mathcal{T})$, $G(f) = \int_{\mathcal{T}} \Sigma(t, t') f(t) dt$. Specifically,

$$\Sigma(t,t') = Cov\big(Y(t), Y(t')\big) = \sum_{k=1}^{\infty} \lambda_k \phi_k(t)\phi_k(t'), (t,t' \in \mathcal{T}). \quad (3.1)$$

Note that the eigenvalues are in a decreasing order such that $\lambda_1 \geq \lambda_2 \geq \cdots 0$, and the eigenfunctions $\phi_k(t)$ serve as orthonormal basis functions in the $L^2(\mathcal{T})$ Hilbert space. Based on equation (3.1), the Karhunen-Loève decomposition of the centered stochastic process $Y(t) - \mu(t)$ can be expressed as:

$$Y(t) - \mu(t) = \sum_{k=1}^{\infty} \xi_k \phi_k(t) + \varepsilon(t), \quad (3.2)$$

where $\xi_k = \int_{\mathcal{T}} \big(Y(t) - \mu(t)\big)\phi_k(t)dt$ is the kth FPC score associated with the kth eigenvalue λ_k and $\varepsilon(t)$ is the additive Gaussian noise. The FPC scores are uncorrelated (i.e., $Cov(\xi_k, \xi_{k'}) = 0, k \neq k'$) with expectation $\mathbb{E}[\xi_k] = 0$ and variance $Var[\xi_k] = \lambda_k$. In practice, the top few FPC scores explain most of the variability in the observed curves, so one can use the approximate decomposition based on the first Q FPC scores:

$$Y(t) \approx \mu(t) + \sum_{k=1}^{Q} \xi_k \phi_k(t) + \varepsilon(t), \quad (3.3)$$

where Q is chosen based on a statistical criterion such as the modified Akaike criterion, Bayesian information criterion, or proportion of explained variance [55]. Next, we elaborate in detail on how to employ FPCA to model the degradation signals (i.e., longitudinal data) without making restrictive parametric assumptions on the degradation trend.

Suppose that there are training data collected from N units, with $\mathcal{I} = \{1,2, \dots, N\}$ denoting the set of training units. Each unit $i \in \mathcal{I}$ has an associated dataset $\boldsymbol{D}_i = [\delta_i, K_i, \boldsymbol{Y}_{i,:}]$, in which $\delta_i = I(F_i \leq C_i)$, $I(\cdot)$ is an indicator function, and $K_i = \min\{F_i, C_i\}$ with its failure time F_i and censoring time C_i. Here, an observation is censored if we do not observe the exact failure time and

we only have observations up to a specific time. Each unit collects degradation signals from J sensors simultaneously and its observed degradation signals are denoted by $\boldsymbol{Y}_{i,:}$:

$$\boldsymbol{Y}_{i,:} = \left(\boldsymbol{Y}_{i,1}, \cdots, \boldsymbol{Y}_{i,J}\right) = \begin{pmatrix} Y_{i,1}\left(t_{i,1}\right) & \cdots & Y_{i,J}\left(t_{i,1}\right) \\ \vdots & \ddots & \vdots \\ Y_{i,1}\left(t_{i,n_i}\right) & \cdots & Y_{i,J}\left(t_{i,n_i}\right) \end{pmatrix} \in \mathbb{R}^{n_i \times J}, \tag{3.4}$$

where $Y_{i,j}(t)$ is the measurement of sensor j in unit i at time t such that $\boldsymbol{Y}_{i,j} = \left[Y_{i,j}\left(t_{i,1}\right), \ldots, Y_{i,j}\left(t_{i,n_i}\right)\right]^T$, n_i represents the number of observations for unit i, and t_{i,n_i} is the n_ith signal observation time of unit i. The degradation signals are also assumed to be observed within a compact time domain $\mathcal{T} = [0, T_{max}]$, where T_{max} can be learned based on domain knowledge or historical degradation signals. For each sensor $j \in \{1, \ldots, J\}$, we apply the FPCA decomposition as follows:

$$Y_{i,j}(t) = \mu_j(t) + X_{i,j}(t) + \varepsilon_{i,j}(t) \approx \mu_j(t) + \sum_{k=1}^{Q_j} \xi_{i,j,k}\, \phi_{j,k}(t) + \varepsilon_{i,j}(t). \tag{3.5}$$

Here, $\mu_j(t)$ is the mean function of sensor j evaluated at time t, $X_{i,j}(t)$ represents the stochastic random deviation from the underlying degradation trajectory, $\xi_{i,j,k}$ is the kth FPC score of sensor j of unit i, $\phi_{j,k}(t)$ is the kth eigenfunction of sensor j at time t, $\varepsilon_{i,j}(t) \sim N(0, \sigma_j^2)$ is the additive Gaussian noise for each sensor j, and Q_j is the number of top FPC scores used to estimate sensor j's signals. Under this FPCA decomposition, the degradation signal $Y_{i,j}(t)$ follows a stochastic process with mean function $\mu_j(t)$ and stochastic deviations $X_{i,j}(t)$ with mean zero and covariance $\Sigma(t, t')$.

3.2.2 Sub-model 2: Bayesian Neural Network-Cox (BNN-Cox) for Time-to-event Data

In this subsection, we introduce the BNN-Cox sub-model for time-to-event data. As illustrated in Figure 3.1, the fitted degradation signals obtained from the FPCA sub-model will be incorporated into the Cox model as time-varying covariates. As a result, throughout this paper, we will use the terms "time-varying covariates" and "degradation signals" interchangeably.

Let $\widehat{\boldsymbol{Y}}_{i,:}(t) = \left[\hat{Y}_{i,1}(t), \dots, \hat{Y}_{i,J}(t)\right]^T \in \mathbb{R}^{J\times 1}$, where $\hat{Y}_{i,j}(t) = \hat{X}_{i,j}(t) + \hat{\mu}_j(t)$, denote the FPCA-fitted multisensor degradation signals. In traditional joint models [56], the log hazard function is assumed to be a linear combination of the fitted signals (i.e., covariates $\widehat{\boldsymbol{Y}}_{i,:}(t)$). In particular, $\widehat{\boldsymbol{Y}}_{i,:}(t)$ are plugged into the linear Cox formula such that:

$$h_i\left(t|\widehat{\boldsymbol{Y}}_{i,:}(t)\right) = h_0(t)\exp\left[\boldsymbol{\omega}^T\widehat{\boldsymbol{Y}}_{i,:}(t)\right], \tag{3.6}$$

where $\boldsymbol{\omega}^T = \left[\omega_1, \dots, \omega_J\right] \in \mathbb{R}^{1\times J}$ represents the Cox regression coefficients of $\widehat{\boldsymbol{Y}}_{i,:}(t)$, $h_0(t)$ is the baseline hazard function, and $h_i\left(t|\widehat{\boldsymbol{Y}}_{i,:}(t)\right)$ is the overall hazard function of unit i. Nevertheless, in practice, the log hazard function frequently has nonlinear relationships with the degradation signals (i.e., covariates).

As reviewed in Section 3.1, several methods have been proposed to relax this linear-risk assumption by using $\widehat{\boldsymbol{Y}}_{i,:}(t)$ as inputs of a deterministic NN [48], [49], [53]. In particular, the NN output, denoted as $g_i(t)$, replaces the linear combination $\boldsymbol{\omega}^T\widehat{\boldsymbol{Y}}_{i,:}(t)$ in equation (3.6) (i.e., $g_i(t) = g\left(\widehat{\boldsymbol{Y}}_{i,:}(t)\right)$), resulting in the following hazard $h_i(t)$ and survival $S_i(t)$ functions:

$$h_i\left(t|\widehat{\boldsymbol{Y}}_{i,:}(t)\right) = h_0(t)\exp[g_i(t)], \tag{3.7}$$

$$S_i\left(t|\widehat{\boldsymbol{Y}}_{i,:}(t)\right) = \exp\left(-\int_0^t h_i(s)ds\right) = \exp\left(-\int_0^t h_0(s)\exp(g_i(s))\,ds\right). \quad (3.8)$$

However, a critical disadvantage of modeling the function g using a deterministic NN is that its output $g_i(t)$ is a fixed-point estimate that does not consider the uncertainties in the NN parameters.

To overcome this limitation, we propose to leverage a BNN that allows uncertainty quantification of the NN parameters and predictions. Compared to conventional deterministic NNs, a BNN provides prevention against overfitting, increased modeling flexibility, and better small sample properties [57]. Specifically, a BNN places a prior distribution over its weight parameters $\boldsymbol{\Omega}_{BNN}$ and uses Bayes' theorem to compute the posterior predictive distribution. While theoretically sound, the main challenge of using a BNN is that the posterior distribution of $\boldsymbol{\Omega}_{BNN}$ is generally intractable. To tackle this issue, we perform variational inference by introducing a tractable approximate variational distribution q, and then minimizing the Kullback-Leibler (KL) divergence between $q(\boldsymbol{\Omega}_{BNN})$ and the posterior distribution of $\boldsymbol{\Omega}_{BNN}$. In practice, variational inference in a BNN is done by using the popular Monte Carlo dropout (MC dropout) [58]. Dropout is a technique normally used to prevent overfitting in training NNs by randomly dropping nodes and their connections during training [59]. Unlike regular dropout where the "dropping" only happens during model training, MC dropout randomly drops nodes and their connections in both training and testing. In other words, based on the trained NN, we randomly drop some of its nodes and connections during testing as well and perform a (stochastic) forward pass. Studies [58], [60] have shown that conducting this forward pass with MC dropout is equivalent to performing variational inference in a BNN.

To better understand how a BNN quantifies uncertainty via MC dropout, we defer the details of BNN training to Section 3.2.3, and suppose that we now have a trained BNN g and a new test

unit r with predicted degradation signals $\widehat{\boldsymbol{Y}}_{r,:}(t) = \left[\hat{Y}_{r,1}(t), \dots, \hat{Y}_{r,J}(t)\right]^T \in \mathbb{R}^{J\times 1}$ from the FPCA sub-model . The approximate predictive distribution can then be calculated by $q\left(g_r(t)\middle|\widehat{\boldsymbol{Y}}_{r,:}(t)\right) = \int p\left(g_r(t)\middle|\widehat{\boldsymbol{Y}}_{r,:}(t), \boldsymbol{\Omega}_{BNN}\right) q(\boldsymbol{\Omega}_{BNN}) d\boldsymbol{\Omega}_{BNN}$. In practice, we empirically estimate the mean and the variance by plugging in $\widehat{\boldsymbol{Y}}_{r,:}(t)$ as inputs to the trained BNN and repeating V stochastic forward passes through the network to obtain V Monte Carlo samples $g_r^{(1)}(t), \dots, g_r^{(v)}(t), \dots, g_r^{(V)}(t)$. Each $g_r^{(v)}(t) \in \mathbb{R}, (v = 1, \dots, V)$ comes from the distribution $p\left(g_r(t)\middle|\widehat{\boldsymbol{Y}}_{r,:}(t), \widehat{\boldsymbol{\Omega}}_{BNN}^{(v)}\right)$, where $\widehat{\boldsymbol{\Omega}}_{BNN}^{(v)}$ is drawn from the approximate variational distribution $q(\boldsymbol{\Omega}_{BNN})$**.** Finally, we use moment-matching as shown in equation (3.9) and equation (3.10) to estimate the mean and variance of $q\left(g_r(t)\middle|\widehat{\boldsymbol{Y}}_{r,:}(t)\right)$ [58]. In practice, we recommend setting the dropout rate as 0.1 or 0.2 based on the comments from the original authors.

$$\mathbb{E}_{q\left(g_r(t)\middle|\widehat{\boldsymbol{Y}}_{r,:}(t)\right)}(g_r(t)) \approx \frac{1}{V}\sum_{v=1}^{V} g_r^{(v)}(t), \tag{3.9}$$

$$Var_{q\left(g_r(t)\middle|\widehat{\boldsymbol{Y}}_{r,:}(t)\right)}(g_r(t)) \approx \frac{1}{V}\sum_{v=1}^{V}\left(g_r^{(v)}(t)\right)^2 - \left(\frac{1}{V}\sum_{v=1}^{V} g_r^{(v)}(t)\right)^2. \tag{3.10}$$

3.2.3 Offline Parameter Estimation

After defining the two sub-models, we discuss how to estimate the model parameters in an offline setting. Let $\boldsymbol{\Omega} = \{\boldsymbol{\mu}(t), \boldsymbol{\sigma}^2, \boldsymbol{\phi}(t), \boldsymbol{\lambda}, \boldsymbol{\Omega}_{BNN}, h_0(t)\}$ denote the unknown parameters of the proposed IUQ model, in which $\boldsymbol{\mu}(t) = \left[\mu_1(t), \dots, \mu_J(t)\right]^T$ is the set of mean functions for each sensor, $\boldsymbol{\sigma}^2 = \left[\sigma_1^2, \dots, \sigma_J^2\right]^T \in \mathbb{R}^{J\times 1}$ are the additive Gaussian error terms for each sensor, $\boldsymbol{\lambda} = \left[\boldsymbol{\lambda}_1; \dots; \boldsymbol{\lambda}_J\right] \in \mathbb{R}^{(\Sigma_{j=1}^{J} Q_j)\times 1}$ is the set of eigenvalues such that $\boldsymbol{\lambda}_j = \left[\lambda_{j,1}, \dots, \lambda_{j,Q_j}\right]^T \in \mathbb{R}^{Q_j\times 1}$, and

$\boldsymbol{\phi}(t) = \left[\boldsymbol{\phi}_1(t), \dots, \boldsymbol{\phi}_J(t)\right]^T$ is the corresponding set of eigenfunctions such that $\boldsymbol{\phi}_j(t) = [\boldsymbol{\phi}_{j,1}(t), \dots, \boldsymbol{\phi}_{j,Q_j}(t)]$. To estimate the model parameters $\boldsymbol{\Omega}$, a natural approach is to maximize the joint likelihood function. In particular, the joint likelihood function $\mathcal{L}(\boldsymbol{D}; \boldsymbol{\Omega})$ can be written as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{D}; \boldsymbol{\Omega}) = p(\boldsymbol{K}, \boldsymbol{\delta}, \boldsymbol{Y}; \boldsymbol{\Omega}) &= \int p(\boldsymbol{K}, \boldsymbol{\delta}, \boldsymbol{Y}|\boldsymbol{\xi}; \boldsymbol{\Omega}) p(\boldsymbol{\xi}; \boldsymbol{\Omega}) d\boldsymbol{\xi} \\ &= \int p(\boldsymbol{K}, \boldsymbol{\delta}|\boldsymbol{\xi}; \boldsymbol{\Omega}) p(\boldsymbol{Y}|\boldsymbol{\xi}; \boldsymbol{\Omega}) p(\boldsymbol{\xi}; \boldsymbol{\Omega}) d\boldsymbol{\xi} \\ &= \int \prod_{i=1}^{N} p(K_i, \delta_i|\boldsymbol{\xi}_i; \boldsymbol{\Omega}) p\left(\boldsymbol{Y}_{i,:}\middle|\boldsymbol{\xi}_i; \boldsymbol{\Omega}\right) p(\boldsymbol{\xi}_i; \boldsymbol{\Omega}) d\boldsymbol{\xi}_i,\end{aligned} \tag{3.11}$$

where $\boldsymbol{K} = [K_1, \dots, K_N]^T \in \mathbb{R}^{N\times 1}$, $\boldsymbol{\delta} = [\delta_1, \dots, \delta_N]^T \in \mathbb{R}^{N\times 1}$, and $\boldsymbol{\xi}_i = \left[\boldsymbol{\xi}_{i,1}; \dots; \boldsymbol{\xi}_{i,J}\right] \in \mathbb{R}^{JQ\times 1}$.

The joint likelihood function in equation (3.11) consists of three parts, where each component corresponds to the BNN-Cox sub-model ($p(K_i, \delta_i|\boldsymbol{\xi}_i; \boldsymbol{\Omega})$), the FPCA sub-model ($p\left(\boldsymbol{Y}_{i,:}\middle|\boldsymbol{\xi}_i; \boldsymbol{\Omega}\right)$), and the prior distribution of the FPC scores ($p(\boldsymbol{\xi}_i; \boldsymbol{\Omega})$). Specifically, the exact form of each component can be written as follows:

$$\begin{aligned}p(K_i, \delta_i|\boldsymbol{\xi}_i; \boldsymbol{\Omega}) &= h_i(K_i)^{\delta_i} S_i(K_i) \\ &= \{h_0(K_i) \exp[g_i(K_i)]\}^{\delta_i} \exp\left\{-\int_0^{K_i} h_0(s) \exp[g_i(s)]\, ds\right\},\end{aligned} \tag{3.12}$$

$$p\left(\boldsymbol{Y}_{i,:}\middle|\boldsymbol{\xi}_i; \boldsymbol{\Omega}\right) = \prod_{j=1}^{J} p\left(\boldsymbol{Y}_{i,j}\middle|\boldsymbol{\xi}_{i,j}; \boldsymbol{\Omega}\right) = \prod_{j=1}^{J} \left(2\pi\sigma_j^2\right)^{-\frac{n_i}{2}} \exp\left\{-\frac{\left\|Y_{i,j} - \hat{Y}_{i,j}\right\|^2}{2\sigma_j^2}\right\}, \tag{3.13}$$

$$p(\boldsymbol{\xi}_i; \boldsymbol{\Omega}) \sim \mathcal{D}, \tag{3.14}$$

where $\mathcal{D}$ represents a general distribution that is selected based on the modeling assumptions.

Directly optimizing the joint likelihood in equation (3.11), e.g., via the Expectation-Maximization algorithm with numerical integration [61] may lead to heavy computational loads and even numerical instability issues due to the high-dimensional integration. To alleviate this

issue, we adopt the "two-stage" approach to sequentially estimate the IUQ model's parameters [62]. Here, we first optimize the FPCA parameters and then sequentially estimate the BNN-Cox parameters. Many existing studies have demonstrated that such a two-stage approach can yield competitive results with negligible bias as directly estimating the joint model [14], [23].

For the FPCA sub-model, we follow the popular approach by [63] where the mean function $\hat{\mu}(t)$ and the covariance function $\hat{\Sigma}_j(t,t')$ are estimated by smoothing methods like local linear smoothing or spline smoothing, and σ_j^2 is estimated by smoothing $Y_{i,j}(t) - \hat{\mu}_j(t)^2 - \hat{\Sigma}_j(t,t)$ against t using a local linear smoother. One point of caution is that local linear smoothers can introduce unwanted bias during the estimation of the mean and covariance functions, especially under signal truncation. In this situation, we can replace the local linear smoothers with penalized splines [64] to mitigate the estimation bias. Also, note that since we assume a random signal truncation scenario based on hard failures, the degree of bias is negligible compared to the soft failure assumption [40].

Finally, the eigen-components are derived by solving the eigen-equations:

$$\int_{\mathcal{T}} \hat{\Sigma}_j(t,t')\,\hat{\phi}_{j,k}(t)dt = \hat{\lambda}_{j,k}\hat{\phi}_{j,k}(t), \tag{3.15}$$

where the eigenfunctions are constrained to satisfy $\int_{\mathcal{T}} \hat{\phi}_{j,k}(t)^2\,dt = 1$ and $\int_{\mathcal{T}} \hat{\phi}_{j,k}(t) \cdot \hat{\phi}_{j,k'}(t) = 0$ for $k < k'$. In practice, deriving the FPC scores based on their definition $\xi_k = \int_{\mathcal{T}} \big(Y(t) - \mu(t)\big)\phi_k dt$ can be challenging when the observed longitudinal data is highly sparse. To overcome this challenge, a widely used approach [63] is to assume that $\xi_{i,j,k}$ follows a Gaussian distribution and then estimate the FPC scores for unit i through the conditional expectation such that $\hat{\xi}_{i,j,k} = \mathbb{E}\big[\xi_{i,j,k}\big|\boldsymbol{Y}_{i,j}\big] = \hat{\lambda}_{j,k}\,\widehat{\boldsymbol{\phi}}_{j,k}^T\,\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{Y}_{i,j}}^{-1}\big(\boldsymbol{Y}_{i,j} - \widehat{\boldsymbol{\mu}}_{i,j}\big)$ where $\widehat{\boldsymbol{\phi}}_{j,k} = \big[\hat{\phi}_{j,k}\big(t_{i,1}\big), \ldots, \hat{\phi}_{j,k}\big(t_{i,n_i}\big)\big]^T$,

$\widehat{\boldsymbol{\mu}}_{i,j} = \left[\hat{\mu}_{i,j}(t_{i,1}), \ldots, \hat{\mu}_{i,j}(t_{i,n_i})\right]^T$, and $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{Y}_{i,j}}$ is the matrix with (a,b)th entry as $\hat{\Sigma}_j(t_{i,a}, t_{i,b})$ for $1 \leq a, b, \leq n_i$.

For the BNN sub-model, we use the following loss function to train the model:

$$\ell_{BNN} = -\sum_{m=1}^{M}\left\{\sum_{i' \in \boldsymbol{H}(\tau_m)} g_{i'}(\tau_m; \boldsymbol{\Omega}_{BNN}) - \sum_{l=1}^{d_m}\left(\sum_{i \in \boldsymbol{\Theta}(\tau_m)} \exp\left(g_i(\tau_m; \boldsymbol{\Omega}_{BNN})\right) - \frac{l-1}{d_m}\sum_{i' \in \boldsymbol{H}(\tau_m)} \exp\left(g_{i'}(\tau_m; \boldsymbol{\Omega}_{BNN})\right)\right) - \chi \sum_{i \in \boldsymbol{\Theta}(\tau_m)} |g_i(\tau_m; \boldsymbol{\Omega}_{BNN})|\right\}. \quad (3.16)$$

Here, the BNN-Cox loss ℓ_{BNN} in equation (3.16) is based on Efron's approximation [65] of the log Cox partial likelihood equation with modifications to support tied event times. In particular, let $\tau_1, \ldots, \tau_M$ be the unique ordered M failure times such that $(\tau_1 < \tau_2 < \cdots < \tau_M)$, $\boldsymbol{\Theta}(\tau_i) = \{j | K_j \geq \tau_i\}$ be the risk set (i.e., units subject to failure at failure time τ_i), $\boldsymbol{H}(\tau_m)$ be the set of units that failed at time τ_m, and d_m be the number of failures at time τ_m. Specifically, the first and second terms in equation (3.16) each represent the log-transformed numerator and the denominator of Efron's approximation, and the last term $\sum_{i \in \boldsymbol{\Theta}(\tau_m)} |g_i(\tau_m)|$ acts as the regularization term to prevent the BNN-Cox from overfitting with χ as the tuning parameter.

The BNN loss ℓ_{BNN} can be optimized using gradient-based methods, and here we use the Adam optimizer with a fixed learning rate to optimize the loss function. Finally, the baseline hazard function $h_0(t)$ is estimated using Breslow's approximation [66] shown below:

$$\hat{h}_0(\tau_m) = \frac{d_m}{\sum_{i \in \boldsymbol{\Theta}(\tau_m)} \exp\left(g_i(K_i)\right)}. \quad (3.17)$$

Since the baseline hazard estimates $\hat{h}_0(\tau_m)$ are random due to the BNN-Cox term $g_i(K_i)$, we use

the mean value $\mathbb{E}\left[\hat{h}_0(\tau_m)\right]$ calculated by averaging MC samples from the BNN-Cox. The averaged baseline hazard function $\mathbb{E}\left[\hat{h}_0(\tau_m)\right]$ is then smoothed using a polynomial spline. Smoothing the baseline hazard function is a common practice in survival analysis to 1) improve prediction accuracy; 2) obtain a smooth, continuous estimate rather than some piecewise constant estimate of the baseline hazard; and 3) mitigate the effect of large measurement errors (i.e., "spikes") in the baseline hazard [67].

3.2.4 Online Updating and Prediction with Uncertainty Quantification

This subsection considers an online setting in which new degradation signals $\boldsymbol{Y}_{r,:}(\boldsymbol{t}_r) = \left[Y_{r,1}(\boldsymbol{t}_r), \dots, Y_{r,J}(\boldsymbol{t}_r)\right]^T \in \mathbb{R}^{n_r \times J}$ are observed in times $\boldsymbol{t}_r = \left[t_{r,1}, \dots, t_{r,n_r}\right]^T$ from an in-service unit r, and then we need to update and predict the RUL for this unit with uncertainty quantification. Recall that the in-service unit r is a new unit that does not belong in the training set (i.e., $r \notin \mathcal{I}$). Based on the newly observed degradation signals, we first update the FPCA parameters $\boldsymbol{\xi}_r$.

Similar to [63], we utilize a Bayesian approach to calculate the posterior distribution of the in-service unit's FPC scores $\boldsymbol{\xi}^*_{r,j} = \left[\xi_{r,j,1}, \dots, \xi_{r,j,Q_j}\right]^T$ based on the newly observed signals $\boldsymbol{Y}_{r,:}(\boldsymbol{t}_r)$ and the prior distribution of the FPC scores. Given the decomposition $Y_{r,j}(t) = \mu_j(t) + \sum_{k=1}^{Q_j} \xi_{r,j,k}\phi_{j,k}(t) + \varepsilon_j(t)$ with prior distribution $\xi_{r,j,k} \sim N\left(0, \lambda_{j,k}\right), k = 1, \dots, Q_j$ and $\varepsilon_j(t) \sim N(0, \sigma_j^2)$, the posterior distribution of the FPC scores $\xi^*_{r,j,k} = P\left(\xi_{r,j,k} \middle| \boldsymbol{Y}_{r,:}(\boldsymbol{t}_r)\right)$ can be derived as:

$$\left[\xi^*_{r,j,1}, \dots, \xi^*_{r,j,Q_j}\right]^T \sim MVN\left(\boldsymbol{\xi}^*_{r,j}, \boldsymbol{\Sigma}^*_j\right), \boldsymbol{\xi}^*_{r,j} = \boldsymbol{\Sigma}^*_j \left(\frac{1}{\sigma_j^2}\Phi_j(\boldsymbol{t}_r)^T\left(\boldsymbol{Y}_{r,:}(\boldsymbol{t}_r) - \hat{\mu}_j(\boldsymbol{t}_r)\right)\right), \quad (3.18)$$

$$\boldsymbol{\Sigma}_j^* = \left(\frac{1}{\sigma_j^2}\Phi_j(\boldsymbol{t}_r)^T\Phi_j(\boldsymbol{t}_r) + \boldsymbol{\Lambda}_j^{-1}\right)^{-1}, \Phi_j(\boldsymbol{t}_r) = \begin{pmatrix} \hat{\phi}_{1,j}(t_{r,1}) & \cdots & \hat{\phi}_{Q_j,j}(t_{r,1}) \\ \vdots & \ddots & \vdots \\ \hat{\phi}_{1,j}(t_{r,n_r}) & \cdots & \hat{\phi}_{Q_j,j}(t_{r,n_r}) \end{pmatrix},$$

$$\boldsymbol{\Lambda}_j = diag\left(\hat{\lambda}_{j,1}, \dots, \hat{\lambda}_{j,Q_j}\right), \hat{\mu}_j(\boldsymbol{t}_r) = \left[\hat{\mu}_j(t_{r,1}), \dots, \hat{\mu}_j(t_{r,n_r})\right]^T.$$

Given the updated posterior FPC scores of the in-service unit r, the predicted signal at time $t \in (t^*, T_{max}]$ can be expressed using the posterior mean as:

$$\hat{Y}_{r,j}(t) = \hat{\mu}_j(t) + \sum_{k=1}^{Q_j} \hat{\xi}_{r,j,k}^* \hat{\phi}_{j,k}(t), \tag{3.19}$$

where t^* denotes the prediction time. Note that the mean function $\hat{\mu}_j(t)$ and the eigenfunctions $\hat{\phi}_{j,k}(t)$ are obtained from the training data.

Next, we calculate the conditional survival function using the predicted signals. In particular, the conditional survival function is defined as the probability of survival conditional on the fact that the unit survives at least up to time $t^* \leq t$. In other words, the degradation trajectory of the signals from prediction time t^* to the desired time t is used as a predictor.

$$S\left(t\middle|t^*, \boldsymbol{\xi}_r^*; \widehat{\boldsymbol{\Omega}}\right) = \frac{S\left(t\middle|\boldsymbol{\xi}_r^*; \widehat{\boldsymbol{\Omega}}\right)}{S\left(t^*\middle|\boldsymbol{\xi}_r^*; \widehat{\boldsymbol{\Omega}}\right)} = \exp\left\{-\int_{t^*}^{t} \hat{h}_0(s)\exp\left[g\left(\boldsymbol{Y}_{r,:}(s)\right)\right]ds\right\}. \tag{3.20}$$

The conditional survival function in equation (3.20) can be marginalized by integrating the estimated FPC scores $\boldsymbol{\xi}_r^*$ out:

$$S\left(t\middle|t^*; \widehat{\boldsymbol{\Omega}}\right) = \int S\left(t\middle|t^*, \boldsymbol{\xi}_r^*; \widehat{\boldsymbol{\Omega}}\right)p\left(\boldsymbol{\xi}_r^*; \widehat{\boldsymbol{\Omega}}\right)d\boldsymbol{\xi}_r^*. \tag{3.21}$$

Equation (3.21) is then approximated through a Monte Carlo integration approach:

$$\hat{S}\left(t\middle|t^*; \widehat{\boldsymbol{\Omega}}\right) = \frac{1}{M}\sum_{m=1}^{M} S\left(t\middle|t^*, \hat{\boldsymbol{\xi}}_r^{*(m)}; \widehat{\boldsymbol{\Omega}}\right),\ \hat{\boldsymbol{\xi}}_r^{*(m)} \sim MVN\left(\boldsymbol{\xi}_{r,j}^*, \boldsymbol{\Sigma}_j^*\right). \tag{3.22}$$

A key challenge in the marginalization procedure in equation (3.21) is that in addition to the uncertainties in the FPC scores $\boldsymbol{\xi}_r^*$, there are uncertainties in the NN parameters which are reflected

by the randomness in the BNN-Cox outputs $g\left(\boldsymbol{Y}_{r,:}(s)\right)$. To overcome this challenge, we first generate $m = 1, \dots, M$ posterior samples of the FPC scores $\hat{\boldsymbol{\xi}}_r^{*(m)}$. Then, we reorganize the $S\left(t\middle|t^*, \hat{\boldsymbol{\xi}}_r^{*(m)}; \widehat{\boldsymbol{\Omega}}\right)$ term in equation (3.22) by leveraging the BNN formulation via MC dropout in equation (3.9).

$$\hat{S}\left(t\middle|t^*; \widehat{\boldsymbol{\Omega}}\right) = \frac{1}{M}\sum_{m=1}^{M} S\left(t\middle|t^*, \hat{\boldsymbol{\xi}}_r^{*(m)}; \widehat{\boldsymbol{\Omega}}\right)$$

$$= \frac{1}{M}\sum_{m=1}^{M} \exp\left\{-\int_{t^*}^{t} \hat{h}_0(s) \exp\left[\mathbb{E}_{q\left(g_r(s)\middle|\boldsymbol{Y}_{r,:}(s)\right)}\left(g_r(s)\right)\right] ds\right\}, \tag{3.23}$$

$$\mathbb{E}_{q\left(g_r(s)\middle|\boldsymbol{Y}_{r,:}(s)\right)}\left(g_r(s)\right) = \frac{1}{V}\sum_{v=1}^{V} g(\widehat{\boldsymbol{Y}}_{r,:}^{*(m)}(s); \boldsymbol{\Omega}_{BNN,1}^{(v)}, \dots, \boldsymbol{\Omega}_{BNN,L}^{(v)}),$$

in which L represents the number of layers in the BNN. In equation (3.23), the marginal survival function $\hat{S}\left(t\middle|t^*; \widehat{\boldsymbol{\Omega}}\right)$ from the proposed IUQ model provides a more comprehensive quantification of the uncertainties from *both* sub-models in an integrative fashion. In particular, the uncertainties in the longitudinal sub-model are accommodated by integrating over $\boldsymbol{\xi}_r^*$, while the uncertainties in the BNN are accounted for by summing over the BNN weight parameters in each layer $\boldsymbol{\Omega}_{BNN,1}^{(v)}, \dots, \boldsymbol{\Omega}_{BNN,L}^{(v)}$. As a result, the IUQ model provides more reliable predictions with complete characterizations of the involved modeling uncertainty.

Finally, the expected RUL can be calculated using the estimated marginal survival function $\hat{S}\left(t\middle|t^*; \widehat{\boldsymbol{\Omega}}\right)$:

$$\widehat{RUL}(t^*) = \int_{t^*}^{\infty} \hat{S}\left(t\middle|t^*; \widehat{\boldsymbol{\Omega}}\right) dt. \tag{3.24}$$

The integration in equation (3.24) is numerically evaluated using the Gauss-Legendre quadrature method [40].

3.3 Evaluation

The proposed IUQ model is evaluated using both simulated and real-life data. Details of the evaluation procedures and benchmark methods are provided below.

3.3.1 Benchmark Methods

The IUQ model is evaluated against the state-of-the-art survival models. Table 3.1 summarizes the benchmark methods based on their ability to incorporate time-varying covariates, to nonparametrically capture complex longitudinal trends, to capture nonlinear relationships between the log hazard and the covariates, and to provide uncertainty quantification of the longitudinal and time-to-event sub-models. First, the simplest linear Cox model [68] is added as a baseline. The next benchmark is the DeepSurv [48] model, which is arguably the most popular NN extension of the Cox model. DeepSurv relaxes the linear-risk assumption between the log hazard function and the covariates by using a feedforward NN. Another benchmark is the PyCox model [49] where the authors improve the computational efficiency of the DeepSurv model by utilizing case-control sampling. Although both DeepSurv and PyCox are more flexible, they do not accommodate time-varying covariates in the modeling and only rely on the latest observation to make RUL predictions. Furthermore, they are incapable of providing modeling uncertainties. Finally, the last benchmark is the recent NN-Joint model [53]. This is the first joint modeling approach that uses an NN-extended Cox model and a mixed-effects model. Unlike DeepSurv and PyCox, the NN-Joint model does include time-varying covariates in the modeling procedure and uncertainty quantification for modeling longitudinal data. However, it suffers from 1) limited modeling flexibility due to the parametric mixed-effects model; and 2) imperfect uncertainty estimates by ignoring the uncertainties from modeling time-to-event data.

Table 3.1 Summary of Benchmark Methods and Their Properties

Model	Time-varying Covariates	Nonparametric (Longitudinal)	Nonlinear Risk	Uncertainty Quantification (Longitudinal)	Uncertainty Quantification (Time-to-event)
Linear Cox	O	X	X	O	X
DeepSurv	X	X	O	X	X
PyCox	X	X	O	X	X
NN-Joint	O	X	O	O	X
IUQ (*Proposed*)	O	O	O	O	O

3.3.2 Simulation Study

The performance of the IUQ model is evaluated under comprehensive simulation studies. We generate synthetic degradation signals for two sensors ($J = 2$) with the following form:

$$Y_{i,j}(t) = \boldsymbol{Z}_j^T(t)\boldsymbol{B}_{i,j} + \epsilon_j(t), \tag{3.25}$$

where $\boldsymbol{Z}_j^T$ are the basis functions for sensor j, $\boldsymbol{B}_{i,j}$ are the corresponding coefficients for unit i, and $\epsilon_j(t) \sim N(0, \sigma_j^2)$ is an additive Gaussian error term.

The detailed simulation procedure is organized into the following steps:

Step 1: Generate $N = 350$ samples of $\boldsymbol{B}_{i,1} \sim MVN(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{B}_{i,2} \sim MVN(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ using the parameters from Table 3.2.

Step 2: Define the true hazard function according to equation (3.27) alongside the baseline hazard function in equation (3.26).

Step 3: Generate failure times F_i for each unit by sampling from its distribution $f_i(t) = h_i(t)S_i(t)$ via rejection sampling.

Step 4: Randomly choose 5% of the N units to be censored, in which the censoring time C_i is sampled from a $Unif(1, F_i)$ distribution.

Step 5: Generate noisy degradation signals by equation (3.25) using $\boldsymbol{B}_{i,1}$ and $\boldsymbol{B}_{i,2}$ from step 1, with an additional Gaussian noise term $\epsilon_j(t)$ for $j = 1,2$.

Step 6: Split simulated units into 300 training units and 50 test units. Observations from the test units are truncated up to a pre-specified prediction time t^*.

Table 3.2 Simulation Study Parameter Settings

Parameters	Sensor 1	Sensor 2
σ_j	0.3	0.3
$\boldsymbol{\mu}_j$	$[2.4, 0.1, 0.001]$	$[1.7, 0.1, 0.001]$
$\boldsymbol{\Sigma}_j$	$\begin{pmatrix} 0.2 & -4e-4 & 6e-5 \\ -4e-4 & 2e-7 & 3e-7 \\ 6e-5 & 3e-7 & 3e-6 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 3e-5 & 4e-5 \\ 3e-5 & 2e-5 & 1e-7 \\ 4e-5 & 1e-7 & 3e-6 \end{pmatrix}$
$\mathbf{Z}_j^T(t)$	$[1, t, t^2]$	$[1, t^{0.7}\sin t\,, t^2]$

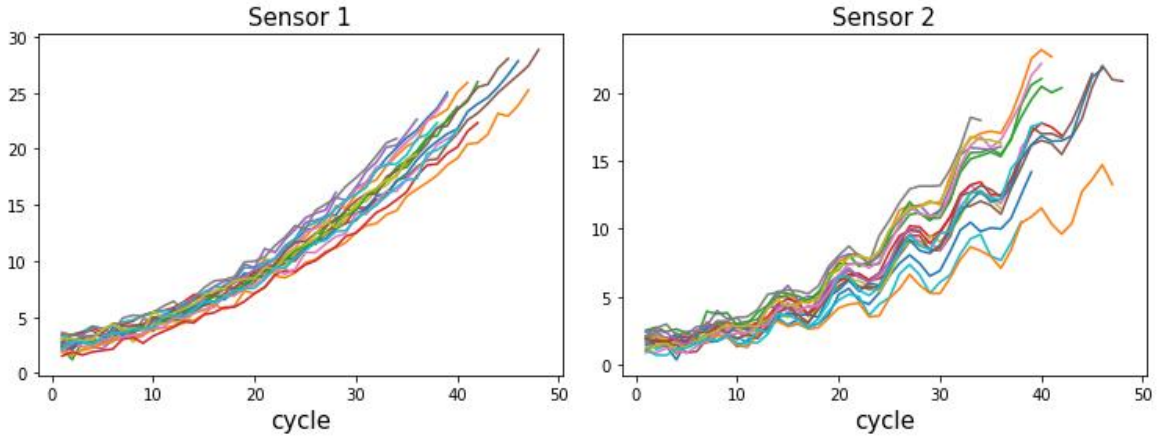
Here, we impose different basis functions for sensor 1 and sensor 2 since degradation signals can have varying trends ranging from cubic, cyclical, piecewise, etc. In particular, we let $\boldsymbol{Z}_1^T(t) = [1, t, t^2]$ be the polynomial basis function for sensor 1 and let $\boldsymbol{Z}_2^T(t) = [1, t^{0.7}\sin t\,, t^2]$ be the custom basis function for sensor 2. $\boldsymbol{B}_{i,j}$ are the random effect coefficients assumed to follow a multivariate normal distribution, i.e., $\boldsymbol{B}_{i,j} \sim MVN(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ The mean of the random effect coefficients is chosen as $\boldsymbol{\mu}_1 = [2.4, 0.1, 0.001]$ and $\boldsymbol{\mu}_2 = [1.7, 0.1, 0.001]$ to impose monotonically increasing degradation trends. Since $\boldsymbol{B}_{i,j}$ follows a Gaussian distribution, it is possible to generate a sample that violates the monotonicity with a very marginal probability. In such cases, we discard the sample and generate a new one to ensure that the underlying degradation process is monotonic. Next, the baseline hazard function is specified according to the Weibull distribution:

$$h_0(t) = \lambda\alpha t^{\alpha-1}, \tag{3.26}$$

where $\alpha = 1.05$ is the shape parameter and $\lambda = 0.0001$ is the scale parameter. The true hazard function of unit i is then defined as:

$$h_i(t) = 10^{-4} \times 1.05 \times t^{1.05-1} \times \exp\left[\left(0.06\left(\boldsymbol{Z}_1^T(t)\boldsymbol{B}_{i,1}\right)^2 + 0.05\left(\boldsymbol{Z}_2^T(t)\boldsymbol{B}_{i,2}\right)^2\right)^{0.5}\right] \quad (3.27)$$

Notice that the true hazard function in equation (3.27) has nonlinear dependencies with the covariates $\boldsymbol{Z}_1^T(t)\boldsymbol{B}_{i,1}$ and $\boldsymbol{Z}_2^T(t)\boldsymbol{B}_{i,2}$. The training units have access to all measurements until failure or censoring. For the test units, we assume that the measurements are available up to a prespecified prediction time t^*, which is smaller than the minimum failure time of the test units. The simulated degradation signals for each sensor are plotted in Figure 3.2. We set $Q_j = 3$ for $j =$



(a) (b)

Figure 3.2 Example of simulated degradation trends of (a) Sensor 1 and (b) Sensor 2 of randomly generated 50 units

1,2.

To better understand the model behavior, we select a random test unit (id: 310) with $\boldsymbol{B}_{i,1} = [2.4436, 0.1030, 0.0131]$ and $\boldsymbol{B}_{i,2} = [1.6542, 0.1013, 0.0094]$ and examine its conditional survival curve and RUL estimates at different prediction times t^*. For instance, $t^* = 5$ implies that the test observations up to time 5 are assumed to be available for online updating, and the survival function is predicted for times greater than $t^* = 5$.

First, we examine the uncertainty quantifications from the proposed IUQ model and the NN-Joint model. Note that the NN-Joint model is selected as the main benchmark as it is the most

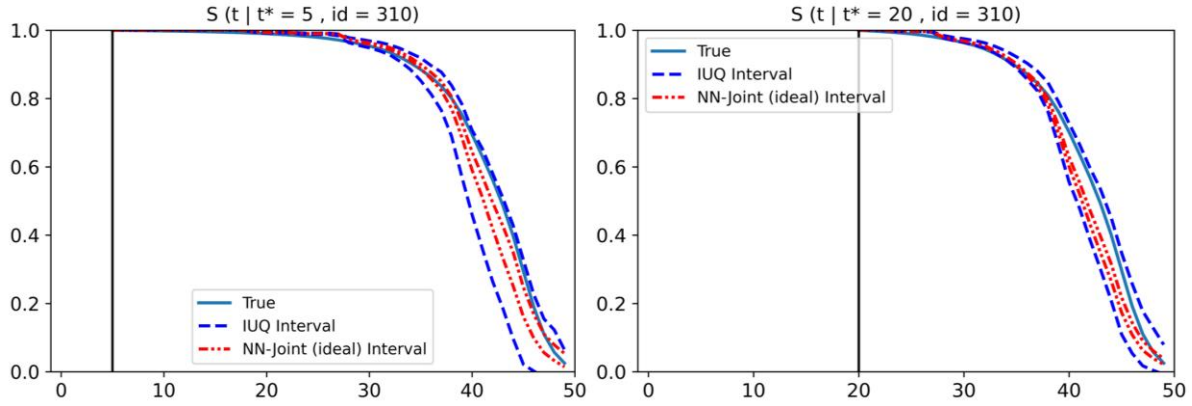


Figure 3.3 Predicted conditional survival curves for $t^* = [5,20]$ in a randomly selected test unit (Green solid: Ground Truth, Blue dashed: ± 3 standard deviations from IUQ mean predictions, Red dash dotted: ± 3 standard deviations from NN-Joint (ideal) mean predictions, Black solid vertical line: t^*)

recent model that utilizes an NN-extended Cox model with the best state-of-the-art performance. To remove the effect of basis functions, we let NN-Joint (ideal) model know the ideal basis functions (i.e., $[1, t, t^2]$ for sensor 1 and $[1, t^{0.7} \sin t\,, t^2]$ for sensor 2. However, this comparison is a little unfair to our IUQ model as IUQ does not know this underlying basis function and it is challenging for the NN-Joint model to know the exact basic functions, or the degradation signals

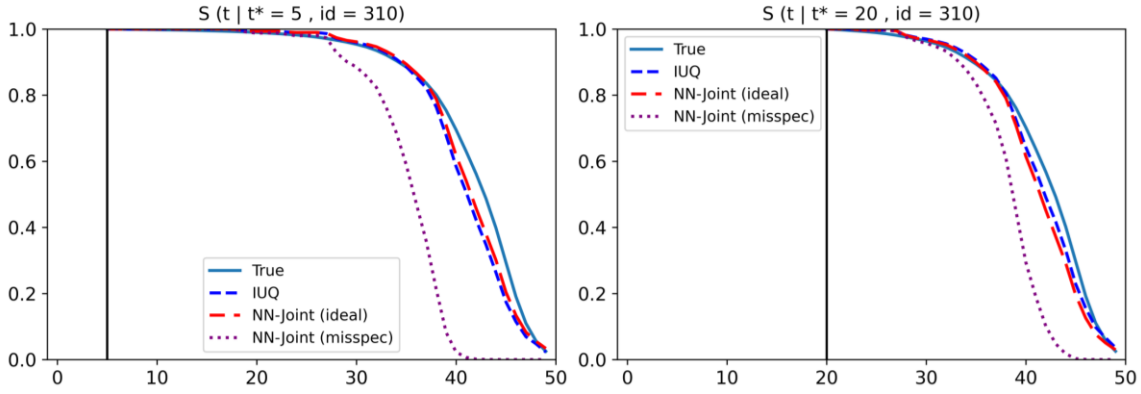


Figure 3.4 Predicted conditional survival curves for $t^* = [5,20]$ in a randomly selected test unit (Green solid: Ground Truth, Blue dashed: IUQ, Red long dashed: NN-Joint (ideal), Purple dotted: NN-Joint (misspec), Black solid vertical line: t^*)

may not even come from such a parametric form in equation (3.25). Figure 3.3 and Figure 3.4 shows the predicted conditional survival curves and their prediction intervals (with ± 3 standard deviations) from both IUQ model and the NN-Joint model. We observe from Figure 3.3 that the IUQ model's prediction intervals cover the true survival curve, whereas the prediction intervals from the NN-Joint model are overconfident with very narrow intervals that do not cover the true survival curve. The overconfidence of the NN-Joint model is likely due to its failure to incorporate the uncertainties of the NN-extended Cox model. Hence, the IUQ model accurately quantifies all sources of modeling uncertainty, while the NN-Joint model fails to do so. Furthermore, the NN-Joint's prediction intervals become more overconfident (i.e., more narrow intervals) at later prediction times ($t^* = 20$) than earlier prediction times ($t^* = 5$). This can be problematic in practice since it is desirable to have more accurate prediction intervals as the unit approaches failure. On the contrary, the IUQ model provides accurate prediction intervals that cover the true survival curve at both early and late prediction times.

Second, we investigate the flexibility of the IUQ model. The functional form of the degradation signals is rarely known in practice, hence parametric models are susceptible to misspecification errors. The IUQ model overcomes this challenge by using a flexible, nonparametric FPCA-based model to infer the functional form of the degradation signals. To highlight the benefits of this added flexibility, we consider two scenarios. The first scenario is the misspecification scenario (NN-Joint (misspec)), where we assume that sensor 1 and sensor 2 both follow a quadratic trend (i.e., $[1, t, t^2]$ for sensors 1 and 2) based on visual inspection. The second scenario (NN-Joint (Ideal)) is the ideal scenario, which assumes that the true functional form of both degradation signals is known. Results from Figure 3.4 show that the NN-Joint model is very sensitive to the choice of basis functions, while the IUQ model is free from this phenomenon due to its

nonparametric approach. Throughout all prediction times, we observe that the misspecified survival curve in purple is drastically different from the true survival curve. On the contrary, the IUQ model's predicted mean survival curves are nearly identical to that of the NN-Joint model with ideal basis functions, suggesting that the IUQ model can accurately capture complex degradation trajectories without relying on prior domain knowledge. Moreover, it is worth noting that despite the similarity in the mean survival predictions, the uncertainty estimates of the IUQ model are significantly more accurate than that of the NN-Joint model. This further underscores the significance of obtaining accurate predictions and uncertainty estimates simultaneously.

Next, we thoroughly evaluate the model across multiple test units. The same evaluation procedure is conducted across 50 simulated test units. The metrics used for the evaluations are defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_{test}} \left(\widehat{RUL}_i - RUL_i\right)^2}{N_{test}}}, MAE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left[\widehat{RUL}_i - RUL_i\right|, \tag{3.28}$$

$$Coverage\ Ratio_i = I\{Lower \leq S_i(t^* + \Delta t | t^*) \leq Upper\},$$
$$Coverage\ Ratio = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} Coverage\ Ratio_i, \tag{3.29}$$

where $N_{test} = 50$ refers to the number of test units, and $\widehat{RUL}_i$ and RUL_i each refers to the predicted and true RUL estimates of unit i. The root mean squared error (RMSE) and mean absolute error (MAE) is computed at different prediction times $t^* = [5,10,15,20]$. The coverage ratio defined in equation (3.29) is used to measure the quality of the prediction intervals. Here, $Lower$ and $Upper$ each represent the lower and upper prediction intervals with ± 3 standard deviations, and Δt denotes the number of time steps into the future that we wish to make predictions. For instance, if $\Delta t = 20$ and $t^* = 5$, we predict the conditional survival curve for the next 20 time steps into the future starting from prediction time 5. For the following evaluations, we set $\Delta t = 20,30$ to evaluate the model's performance at later stages in time close to failure. To

have an accurate comparison, the evaluations are repeated 50 times.

Table 3.3 RUL Prediction Results for Simulation Study

Metric	t^*	PyCox	DeepSurv	Linear Cox (Misspec)	Linear Cox (Ideal)	NN-Joint (Misspec)	NN-Joint (Ideal)	IUQ (Proposed)
MAE	5	11.718	7.533	6.581	4.446	5.450	**1.709**	1.749
	10	11.492	7.529	5.392	4.240	4.742	1.696	**1.683**
	15	10.704	7.340	4.812	3.689	4.097	1.636	**1.635**
	20	9.284	6.825	4.281	3.610	2.797	1.591	**1.587**
RMSE	5	11.899	7.817	6.899	4.906	5.832	2.213	**2.206**
	10	11.676	7.818	5.769	4.691	5.166	**2.163**	2.178
	15	10.899	7.627	5.220	4.195	4.557	**2.114**	2.128
	20	9.514	7.140	4.750	4.105	3.335	**2.083**	2.127

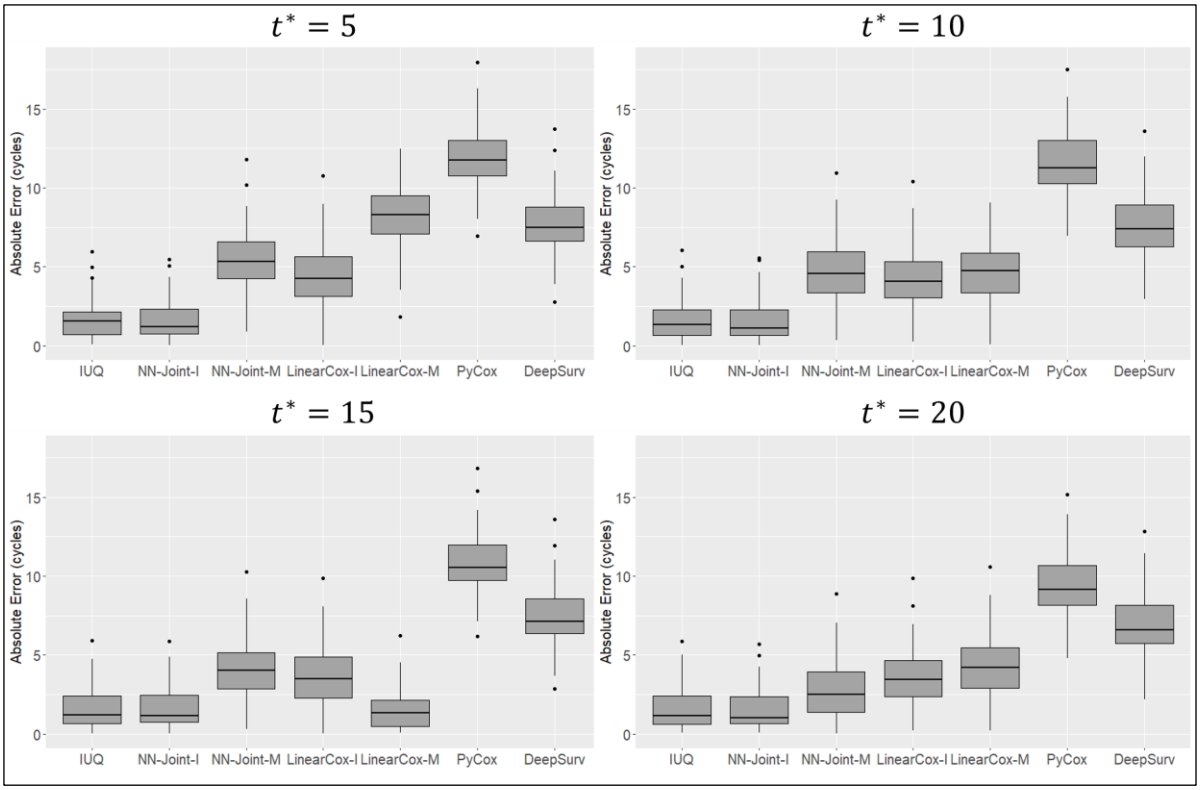


Figure 3.5 RUL prediction accuracy at different prediction times ("NN-Joint-I" refers to the NN-Joint model under ideal basis functions, while "NN-Joint-M" refers to the NN-Joint model under misspecified basis functions. Similarly, "LinearCox-I" means the linear Cox model under ideal basis functions and "LinearCox-M" means the linear Cox model under misspecified basis functions.)

Table 3.3 shows the RUL prediction results, while Figure 3.5 shows the boxplots of the absolute errors measured across repeated evaluations. The proposed model again maintains competitive

prediction results compared to the state-of-the-art NN-Joint model with ideal basis functions. Similar to the previous analysis, the proposed model makes accurate predictions even in earlier prediction times (i.e., $t^* = 5$ in Table 3.3) and that its predictions are more reliable (i.e., narrower boxplots) compared to other benchmarks. Both DeepSurv and PyCox models perform the worst across all scenarios. This is expected since these approaches do not consider time-varying covariates, so they can only rely on the latest observation to make RUL predictions. Furthermore, these methods do not allow real-time updating, so the prediction accuracy does not necessarily improve with more observations. The linear Cox model performs next to the NN-Joint model across all categories, with the main limitation being its inability to capture nonlinear relationships present in equation (3.27). As shown in the linear Cox and NN-Joint model errors in Table 3.3, misspecified basis functions can incur significant prediction errors.

Table 3.4 Coverage Ratios for Simulation Study ($\Delta t = 20$)

t^*	IUQ (Proposed)	NN-Joint (Ideal)	Linear Cox (Ideal)
5	**0.8560**	0.3587	0.0024
10	**0.9467**	0.4160	0.0027
15	**0.9200**	0.3747	0.0212
20	**0.9347**	0.4387	0.0600

The coverage ratios of the proposed IUQ model are then compared to that of the NN-Joint model and linear Cox model with ideal basis functions. Note that results for the DeepSurv and PyCox models are not available as both models do not consider uncertainty quantification. The results in Table 3.4 and Table 3.5 show that the proposed IUQ model drastically outperforms all other benchmarks in terms of coverage ratio. Note that the boldfaced entries represent the best

Table 3.5 Coverage Ratios for Simulation Study ($\Delta t = 30$)

t^*	IUQ (Proposed)	NN-Joint (Ideal)	Linear Cox (Ideal)
5	**0.9267**	0.3933	0.1453
10	**0.9773**	0.5160	0.1173
15	**0.9573**	0.4830	0.1253
20	**0.9627**	0.5547	0.1773

coverage ratios across different prediction times.

Again, the IUQ model achieves drastically higher coverage ratios than other benchmark methods. In other words, the IUQ model leads to much more informative and accurate uncertainty quantifications. On the contrary, both NN-Joint and linear Cox models only consider uncertainties in the longitudinal sub-model, resulting in overconfident prediction intervals that fail to cover the true survival curve.

Table 3.6 Average Computational Time for All Methods (In Seconds)

Model	Linear Cox	DeepSurv	PyCox	NN-Joint	IUQ (Proposed)
Training	0.0740	3.3749	3.2712	10.322	10.607
Prediction (1 unit)	0.2567	0.0893	0.0708	0.4025	0.8474

Finally, we discuss hyperparameter optimization and computation time for the simulation study. All the computations are conducted with a 2.50GHz Quad-Core Intel® i5-10300H CPU with 16GB of RAM. For the hyperparameters, the FPCA sub-model does not require much hyperparameter optimization due to its nonparametric nature. For the BNN sub-model, we use cross-validation to determine the optimal structure. Here, "optimal" is defined as the model structure that results in the highest predictive accuracy in terms of mean RUL based on cross-valuation results. The obtained optimal hyperparameter settings are listed in Table 3.10 in Section 3.5.

The computation time for model training and real-time prediction is listed in Table 3.6. The prediction time is the time needed to obtain the RUL estimate for 1 unit. It is worth noting that training times for both NN-Joint and IUQ are almost identical. During the prediction stage, we found that naïve implementation of the IUQ model will lead to longer computational times (3.1923 seconds) as one needs to perform numerical integration MV times. In circumstances where fast online prediction is critical, computational time can be reduced by limiting the times we perform numerical integration. It should be noted that obtaining samples from the posterior FPCA

distribution in equation (3.22) is relatively straightforward due to the closed-form expression. Similarly, the sampling process using MC dropout is also computationally efficient as it is easily parallelizable across multiple processors. In particular, we initially follow the same process described in Section 3.2.4 and retrieve MV MC dropout samples of the log-hazard function (i.e., $g(\widehat{\boldsymbol{Y}}^{*\ (m)}_{r,:}(s); \boldsymbol{\Omega}^{(v)}_{BNN,1}, \dots, \boldsymbol{\Omega}^{(v)}_{BNN,L})$ in equation (3.23)). Then, instead of performing integration on all MV samples, we first calculate the mean and 3 standard deviation limits of the log-hazard samples and then integrate them. This approach is viable since the variations in the MC dropout samples fully characterize the variations in the survival estimates (i.e., $\hat{h}_0(s)$ is fixed). This approach significantly reduces the computational time during online prediction (0.8474 seconds) by minimizing the number of numerical integration steps.

Table 3.7 Average Computational Time with Varying M (In Seconds)

	$M = 20$	$M = 50$	$M = 100$
Prediction (1 unit)	0.8474	2.7869	5.0401

Another important hyperparameter that greatly affects the computation time and accuracy of the IUQ model is the number of MC dropout samples V and the number of posterior samples from the FPCA sub-model denoted by M. Both hyperparameters control the uncertainty integration approach in equation (3.23), which is a critical component of the IUQ model. Since increasing the number of samples lead to longer computation times, we perform additional studies to determine values for V and M that will lead to accurate predictions with reasonably fast computation time. For the number of MC dropout samples V, relevant literature recommends between $V = 10$ to $V = 100$ to estimate the uncertainty [58]. Here, we choose $V = 30$ after preliminary inspection. For the number of posterior FPCA samples M, we measure the predictive performance on the same 50 test units with varying $M = 20,50,100$. The results in Table 3.7 and Table 3.8 show that increasing M does lead to better predictive performance. However, the performance gain from higher M is

marginal despite the substantially longer computation time. Hence, we used $M = 20$ for all evaluations in the simulation study.

Table 3.8 RUL Prediction Results with Varying M

	MAE			RMSE		
t^*	M=20	M=50	M=100	M=20	M=50	M=100
5	1.749	1.748	1.748	2.206	2.205	2.204
10	1.683	1.675	1.675	2.178	2.169	2.169
15	1.635	1.636	1.634	2.128	2.130	2.128
20	1.587	1.588	1.587	2.127	2.130	2.127

3.3.3 Case Study

In this section, we use a real dataset from a study on automotive lead acid battery aging test [69]. The dataset is collected from an accelerated aging test according to the aging cycles defined by the standards in SAE J2801 [70]. The resistance of the batteries (in milliohms) is tracked until the failure event, which is defined as when the battery fails to start the engine of the automobile. The resistance information of each battery is recorded in weekly intervals. A plot of the resistance trajectories of 14 units is shown in Figure 3.6.

There is no known physical relationship between the resistance path of a lead acid battery with respect to time. Following the previous literature [40], both the benchmark methods, the linear Cox model and the Joint-NN model that require predefined basis functions assume that the resistance follows a quadratic degradation trend.

Since the true conditional survival probabilities are unknown, we only evaluate the mean residual life (MRL) based on equation (3.24) and compare it with the true time-to-failure. Similar to the synthetic data, we impose different prediction times with $t^* = [6,9]$. Note that week 6 and week 9 each correspond to roughly 50% and 75% percentiles of the time horizon of this study. The number of MC dropout samples is $V = 30$, and the number of posterior draws from the FPCA sub-model is still set as $M = 20$, and $Q = 3$.Unlike the simulation study, the coverage ratios cannot be

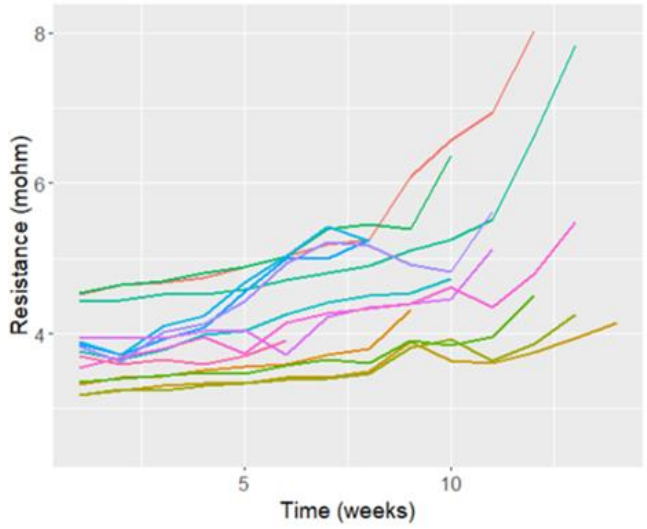
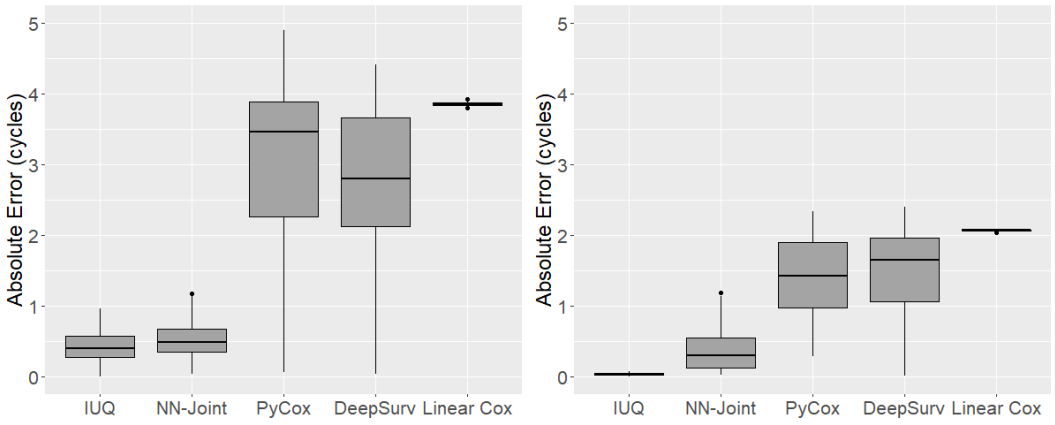


Figure 3.6 Visualization of the Resistance Trajectories of 14 Automotive Lead Acid Batteries

computed since the true survival function is unavailable. Instead, we track the RUL prediction errors of the IUQ model with three variants: 1) IUQ (both) which considers both types of uncertainties; 2) IUQ (longitudinal) which considers uncertainties in the longitudinal sub-model only; and 3) IUQ (time-to-event) which considers uncertainties in the time-to-event sub-model only. We select one battery at random as the test unit and the remaining 13 batteries are used as the training units. To accurately assess model performance, the evaluations are repeated 50 times. The results of the evaluations are summarized in Figure 3.7 and Table 3.9, with the box plot representing the average prediction errors for each unit across the 50 evaluations.

From the case study, we can draw similar conclusions to those of the simulation study. First, the proposed model provides reliable results across varying prediction times. Results show that the proposed model consistently outperforms the existing benchmark methods. Second, the evaluation results from Table 3.9 illustrate the benefits of considering both types of uncertainties in the joint model. Here, the best-performing setup is boldfaced for visual clarity. We observe that incorporating both types of uncertainty results in the most accurate predictions. In addition, the

standard deviation of the errors (shown in parenthesis) is smallest across all scenarios, suggesting that the IUQ model with both types of uncertainties provides the most reliable predictions. Third, the proposed model provides more accurate results as the unit approaches failure (i.e., t^* increases



(a) (b)

Figure 3.7 Evaluation Results on Lead Acid Battery Data at (a) $t^* = 6$ (b) $t^* = 9$

Table 3.9 RUL Prediction Performance for Case Study (MAE)

t^*	IUQ (Both)	IUQ (Longitudinal)	IUQ (Time-to-event)
6	**0.4242** (0.2553)	0.5013 (0.3884)	0.5762 (0.3117)
9	**0.0312** (0.0245)	0.0389 (0.0366)	0.0454 (0.0294)

or more data are collected from the testing unit). In particular, we observe that the prediction gap between the proposed IUQ model and the leading benchmark NN-Joint is more significant at higher prediction times (i.e., $t^* = 9$). While the NN-Joint model barely improves with the newly available measurements, the IUQ's estimates rapidly improve and uncertainty greatly reduces as more test measurements become available for updating. Fourth, our proposed model provides competitive results without any parametric assumptions. The nonparametric modeling approach of the IUQ model effectively captures the complex trends of the degradation signals.

For the linear Cox model, the strong model assumptions (e.g., linear-risk assumption and

misspecified basis functions) again limit the expressiveness of the model. PyCox and DeepSurv models again fall short compared to NN-Joint and the proposed model due to their inability to incorporate covariate history via time-varying covariates and quantify the uncertainties involved.

3.4 Conclusion

This study presented a flexible, accurate, and robust prognostic framework for the joint analysis of longitudinal data and time-to-event data. In particular, an FPCA-based sub-model is used for the longitudinal data, while the BNN-Cox sub-model is employed for the time-to-event data. This study proposes a two-stage inference method to ensure computational efficiency and a Bayesian updating approach to allow real-time RUL prediction of the in-service unit. A major obstacle of existing joint models is that they simply ignore the uncertainties from either sub-model or both entirely. As a result, existing approaches do not provide a comprehensive uncertainty quantification of the survival and RUL estimates. The proposed IUQ model overcomes this challenge by providing an integrated approach for uncertainty quantification. In particular, the IUQ model integrates uncertainties from the longitudinal sub-model (i.e., FPCA) to the time-to-event sub-model (i.e., BNN-Cox), resulting in a more comprehensive characterization of the modeling uncertainty. Second, the flexibility of FPCA and BNN-Cox allows the proposed model to capture complex degradation signal trajectories and covariate interactions. Finally, the proposed method performs well under limited data availability (i.e., censoring). This trait is very useful in practice considering that degradation signals are often truncated or censored due to long development times. Extensive evaluations on synthetic data and real-life battery data demonstrate that the proposed model achieves outstanding and reliable prediction results with accurate prediction intervals.

There are some areas for possible future work. For example, the proposed method assumes that

the degradation signals are collected under a single operational condition and failure mode. In practice, an engineering system can operate across multiple operational conditions with several possible failure modes. To overcome this limitation, we can formulate a competing risks scenario with failure mode-specific hazard function and use operational condition parameters as covariates in the Cox model. Furthermore, the proposed model assumes that the degradation signals are stationary, but the model can be further extended to handle non-stationary signals that exhibit changing behavior over time. One possibility is to reflect the changes in the signals on the FPC scores by repeating the Bayesian updating procedure (introduced in Section 3.2.4) for the first sub-model with FPCA. This updating procedure can be further optimized by incorporating online changepoint detection methods [71] to automatically detect any shifts in the degradation signals. Once a changepoint is detected, it could be used as an indicator to perform the aforementioned Bayesian updating procedure.

3.5 Appendix: Hyperparameter Settings

Table 3.10 Hyperparameter Settings

Model	SIMULATION STUDY	CASE STUDY
Linear Cox	None	None
DeepSurv & PyCox	Hidden layers: 2 Hidden nodes: [32,16] Learning rate: 0.0001 Dropout probability: 0.1 Activation: ReLU Epochs: 100 Batch size: 32	Hidden layers: 2 Hidden nodes: [16,8] Learning rate: 0.0001 Dropout probability: 0.1 Activation: ReLU Epochs: 50 Batch size: 32
NN-Joint	Hidden layers: 2 Hidden nodes: [40,20] Learning rate: 0.0001 Dropout probability: 0.1 Activation: ReLU Epochs: 100 Batch size: 16 Regularization parameter χ: 0.01	Hidden layers: 2 Hidden nodes: [10,10] Learning rate: 0.0001 Dropout probability: 0.1 Activation: ReLU Epochs: 100 Batch size: 8 Regularization parameter χ: 0.01
IUQ (*Proposed*)	Hidden layers: 2 Hidden nodes: [32,16] Learning rate: 0.0001 Dropout probability: 0.1 Activation: ReLU Epochs: 100 Batch size: 16 Regularization parameter χ: 0.01	Hidden layers: 2 Hidden nodes: [10,10] Learning rate: 0.0001 Dropout probability: 0.1 Activation: ReLU Epochs: 100 Batch size: 8 Regularization parameter χ: 0.01

Chapter 4 A Bayesian Spike-and-Slab Sensor Selection Approach for High-dimensional Prognostics

4.1 Introduction

Degradation modeling and prognostics have become increasingly important for improving the economic viability, reliability, and functionality of complex engineering systems. Initial work in this field focused on analyzing a single sensor signal [1] to assess system performance. The underlying assumption of these works is that a single sensor signal is sufficient for characterizing the overall degradation process. However, this assumption is hard to satisfy in modern systems, in which multiple sensors are used to simultaneously monitor various aspects of the system.

Researchers have proposed several data-driven methods to extract prognostic insights from multisensor signals, recognizing that a single sensor is insufficient for fully characterizing the degradation process. These methods include traditional statistical approaches like state-space based models [72], [73], data fusion approaches including health index models [4], [74], [75], and machine learning and deep learning inspired models that leverage the predictive power of neural networks [10], [76], [77]. These methods generally take the multisensor signals as model inputs to predict the remaining useful life (RUL).

One unique and longstanding challenge of analyzing multisensor systems is that each sensor may have varying degrees of relevance to the underlying degradation process [75]. In other words, it is possible that some sensors provide strong insights on the underlying degradation process (i.e., “informative” sensors), while some sensors do not provide insights and just act as noise (i.e., “uninformative” sensors). These uninformative sensors can significantly damage the system’s

overall reliability by compromising the accuracy of RUL predictions. For instance, sensors in wearable devices are commonly used to collect symptom measurements to assess and monitor patient health and progression. However, prior research has shown that the accuracy of monitoring methods is highly reliant on effectively separating the informative sensors from the pool of multisensor measurements [78]. Improper sensor selection may lead to misleading, biased, and non-reproducible results that can potentially harm patient health and prediction [79].

Indeed, the sensor selection challenge is complicated by the widespread use of multiple sensors in various engineering systems. Advances in modern sensor technology have made it practical to adopt numerous sensors to monitor various aspects of the system. For instance, a modern car has on average around 60 to 100 sensors that monitor engine performance, safety features, driver assistance, and other comfort features [80]. As a result, the sensor signals collected from these modern systems are often high-dimensional, meaning that the number of sensors being monitored is much larger than that of traditional systems. Since there are more sensors to select from, the sensor selection challenge becomes more difficult and computationally intensive [81]. Furthermore, in some cases, monitoring all signals is not always viable due to the limited bandwidth or processing capacity. Thus, it is essential to select and only monitor the informative sensors for further prognostic analysis.

Existing methods that tackle the sensor selection challenge in the context of prognostics can be mainly classified into three categories. The first type of approach is heuristic methods, which rely on heuristic rules and visual inspection to identify informative sensors. For instance, Liu et al. [75] removed sensors without consistent monotonic (i.e., decreasing/increasing) trends. This approach is subjective in nature and can vary greatly from one user to another. In addition, this approach does not scale well with the number of sensors, especially in high-dimensional settings.

The second type of approaches is statistics-based approaches. These methods generally involve penalizing the likelihood function by a regularization term to induce sparse solutions. Some examples include the popular least absolute shrinkage and selection operator (LASSO) [82], adaptive LASSO [83], smoothly clipped absolute deviation (SCAD) [84], minmax concave penalty (MCP) [85], and variational inference methods. These methods have already seen success in a wide range of applications, including medical, finance, natural sciences, and healthcare applications. Indeed, researchers have also tried to replicate the success of such methods in prognostics by applying them to address the sensor selection challenge. For instance, Fang et al. [86] first extracted useful features using functional principal components analysis (FPCA) and then applied penalized regression to select informative sensors. Kim et al. [4] employed an adaptive LASSO algorithm with a scaled version of sensor fusion coefficients as the penalty weights. Although these methods are relatively easy to implement, they are known to suffer from estimation bias and provide poor selection results in high-dimensional settings [87].

The last type are deep-learning approaches, which utilize the predictive power of neural networks to automatically identify informative sensors or useful features. For instance, Yu et al. [88] proposed using convolutional gated recurrent units to learn the features of the process data and then used an attention module to preserve the effective features. Another work by Kim et al. [89] proposed a Rectified Linear Unit [90] (ReLU)-based sensor selection network that can be used in conjunction with different neural network-based prognostic models. Although this method showed promising results, it overly limits the flexibility of the neural network in pursuit of interpretability. Furthermore, training this network is unreliable as it frequently falls into local extrema. In addition, none of the above approaches thoroughly investigate sensor selection performance in high-dimensional scenarios with potentially correlated sensors, indicating a

significant gap that requires further study.

To fill the literature gap, this paper investigates the sensor selection challenge to enhance prognostics in high-dimensional settings on the basis of HI-based methods. The main idea of HI-based methods is to construct a 1-D HI by directly combining multiple sensor signals. The constructed 1-D HI can then be used to better characterize the underlying degradation process. Compared to other prognostic methods, HI-based methods provide the following unique benefits. First, the constructed HI provides a real-time visualization and characterization of the underlying degradation evolution, which is much more interpretable than black-box models such as neural networks. This feature is highly sought after in practice by maintenance operators. Second, analyzing a HI has been shown to be more effective in RUL prediction than analyzing the progression of a single sensor [75]. Furthermore, the constructed HI can be regarded as an additional sensor signal that provides a better characterization about the degradation process. Finally, the HI lays a foundation for further prescriptive analysis, which help practitioners make well-informed maintenance decisions. Many works have proposed methods to construct informative HI [4], [75], [91], [92]; however, none of them have addressed the challenges of HI construction and sensor selection in high-dimensional scenarios, which are increasingly common in various industrial applications.

This paper proposes a novel Bayesian spike-and-slab approach for sensor selection and data fusion in high-dimensional settings. In this context, the high-dimensional settings of interest can arise from two scenarios: 1) a low number of training units; and 2) a high number of sensors. In particular, the proposed method simultaneously selects informative sensors and fuses them into a 1-D health index (HI) for further prognostic analysis and RUL prediction. The new contributions of this work are as follows. First, the proposed spike-and-slab sensor selection approach boasts

superior sensor selection performance in high-dimensional scenarios. Second, the proposed approach achieves consistent sensor selection results in the presence of sensor correlation. Third, the proposed approach has desirable theoretical properties such as weak and strong selection consistency. Finally, the proposed method leads to higher RUL prediction accuracy across a wide range of simulation and case studies.

The rest of this paper is organized as follows. Section 4.2 describes how the proposed method selects informative sensors, fuses them into a 1-D HI, and uses the constructed HI to predict RUL. Theoretical properties of the proposed method are also investigated to ensure the sensor selection consistency. Section 4.3 shows the simulation study results to demonstrate the effectiveness of the proposed sensor selection method under varying levels of correlation. Section 4.4 further evaluates the proposed method in a data set of aircraft gas turbine engines and compares it with existing benchmark methods. Finally, Section 4.5 summarizes the key findings and discusses future potential research directions.

4.2 Methodology

In this section, we introduce the proposed Bayesian sensor selection and data fusion method in detail. In Section 4.2.1, we first describe the problem formulation. In Section 4.2.2, we delve into the details of the proposed spike-and-slab priors for sensor selection and elaborate on how to estimate the model parameters. Section 4.2.3 further investigates the theoretical properties of the proposed Bayesian sensor selection approach. Finally, Section 4.2.4 describes how the proposed method predicts the RUL using the constructed HI.

4.2.1 Problem Formulation

Following the recent line of work on health index-based approaches [4], we first define the underlying degradation status and the failure mechanism. Let $\eta_i(t)$ represent the underlying degradation status of unit i at time t. Then, the failure time T_i of unit i is the time when the underlying degradation status $\eta_i(t)$ first reaches the failure threshold l:

$$T_i = \arg\min_t \eta_i(t) \geq l\,. \tag{4.1}$$

Note that unlike previous approaches [4], [74], [75], here we consider a more general setting and do not define the specific form of $\eta_i(t)$. Let $\boldsymbol{L}_i(t) = \left[L_{i,1}(t), \dots, L_{i,s}(t)\right] \in \mathbb{R}^{1\times s}$ denote the measurements of s sensors of unit i at time t. Then, the corresponding HI of unit i at time t, denoted by $h_i(t)$, is defined as such:

$$h_i(t) = z\big(\boldsymbol{L}_i(t)\big) = \eta_i(t) + \varepsilon_i(t), \tag{4.2}$$

in which $z(\cdot)$ is a data fusion function used to recover the underlying degradation status of a unit with the contamination of a Gaussian noise $\varepsilon_i(t) \sim \mathcal{N}(0, \sigma^2)$. Without loss of generality, we set $z(\cdot)$ to be a linear fusion function such that:

$$z\big(\boldsymbol{L}_i(t)\big) = \boldsymbol{L}_i(t)\boldsymbol{w}, \tag{4.3}$$

where $\boldsymbol{w} = [w_1, \dots, w_s]^T \in \mathbb{R}^{s\times 1}$ is the weight vector (i.e., fusion coefficients) to fuse the multisensor signals $\boldsymbol{L}_i(t)$. Note that if one wishes to use a nonlinear fusion function to characterize the degradation process, a linear approximation with K basis functions can be employed such that $z\big(\boldsymbol{L}_i(t)\big) \approx \sum_{k=1}^{K} B_k\big(\boldsymbol{L}_i(t)\big)w_k$. Here, B_k denotes the basis function for $k = 1, \dots, K$, and $B_k\big(\boldsymbol{L}_i(t)\big)$ represents the transformed multisensor signals.

In summary, the relationship between the HI $h_i(t)$, the sensor signals $\boldsymbol{L}_i(t)$, the fusion coefficients $\boldsymbol{w}$, and the underlying degradation status is

$$h_i(t) = \boldsymbol{L}_i(t)\boldsymbol{w} = \eta_i(t) + \varepsilon_i(t). \tag{4.4}$$

For each unit i, the expression above can be rewritten into a matrix form such that $\boldsymbol{L}_i = \left[\boldsymbol{L}_i(t_{i,1}), \dots, \boldsymbol{L}_i(t_{i,n_i})\right]^T \in \mathbb{R}^{n_i \times s}$, $\boldsymbol{\varepsilon}_i = \left[\varepsilon_i(t_{i,1}), \dots, \varepsilon_i(t_{i,n_i})\right]^T \in \mathbb{R}^{n_i \times 1}$, $\boldsymbol{h}_i = \left[h_i(t_{i,1}), \dots, h_i(t_{i,n_i})\right]^T \in \mathbb{R}^{n_i \times 1}$ is the HI vector for unit i, $\boldsymbol{\eta}_i = \left[\eta_i(t_{i,1}), \dots, \eta_i(t_{i,n_i})\right]^T \in \mathbb{R}^{n_i \times 1}$ is the vector of underlying degradation status for unit i, and n_i is the number of sensor measurements from unit i. Hence, we can rewrite equation (4.4) in the following matrix form:

$$\boldsymbol{h}_i = \boldsymbol{L}_i\boldsymbol{w} = \boldsymbol{\eta}_i + \boldsymbol{\varepsilon}_i. \tag{4.5}$$

Our objective is to estimate the fusion coefficients $\boldsymbol{w}$ from the multisensor data of N historical units while only selecting the informative sensors. To distinguish between the informative and uninformative sensors, we first define a set of latent binary indicator variables $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_s] \in \mathbb{R}^{1 \times s}$ for the fusion coefficients $\boldsymbol{w}$ such that $\gamma_j = 1$ if sensor j is included in the HI construction, and $\gamma_j = 0$ if sensor j is excluded from the HI construction. The binary indicators $\boldsymbol{\gamma}$ are additional parameters that need to be estimated alongside the fusion coefficients $\boldsymbol{w}$. In the following subsection, we will discuss how to estimate the parameters of the proposed HI approach.

4.2.2 Bayesian Parameter Estimation

The main parameters that need to be estimated are the fusion coefficients $\boldsymbol{w}$, the latent binary indicators $\boldsymbol{\gamma}$, and the noise variance parameter σ^2. One possible approach is to use the maximum likelihood-based techniques, but these methods are known to suffer from high bias under high dimensions [87]. Instead, we adopt a Bayesian parameter estimation approach by first imposing

carefully designed prior distributions and then obtaining the posterior distribution of the parameters via Bayes' rule.

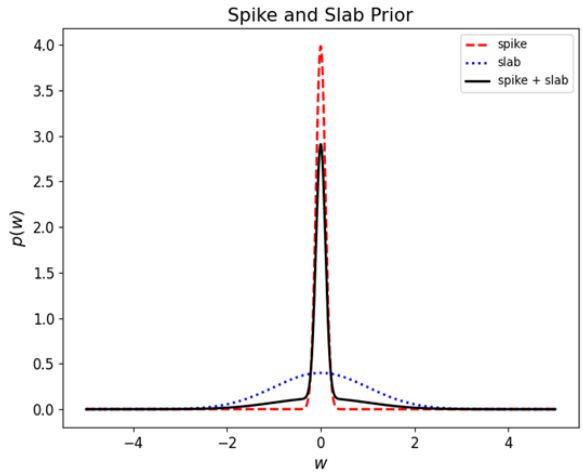


Figure 4.1 Illustration of the Spike and Slab Prior Distribution (Red Dashed Line: Spike, Blue Dotted Line: Slab, Black Solid Line: Spike and Slab)

In particular, we impose the spike-and-slab Gaussian priors that are specifically designed for variable selection in high-dimensional scenarios. Based on equation (4.5), we concatenate the observations from N historical units such that $\boldsymbol{h} = [\boldsymbol{h}_1; \boldsymbol{h}_2; \dots; \boldsymbol{h}_N] \in \mathbb{R}^{\sum_{i=1}^{N} n_i \times 1}$, $\boldsymbol{L} = [\boldsymbol{L}_1; \boldsymbol{L}_2; \dots; \boldsymbol{L}_N] \in \mathbb{R}^{\sum_{i=1}^{N} n_i \times s}$, $\boldsymbol{\eta} = [\boldsymbol{\eta}_1; \boldsymbol{\eta}_2; \dots; \boldsymbol{\eta}_N] \in \mathbb{R}^{\sum_{i=1}^{N} n_i \times 1}$, $\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}_1; \boldsymbol{\varepsilon}_2; \dots; \boldsymbol{\varepsilon}_N] \in \mathbb{R}^{\sum_{i=1}^{N} n_i \times 1}$ and rewrite the sensor signals from all N historical units as:

$$\boldsymbol{h} = \boldsymbol{L}\boldsymbol{w} = \boldsymbol{\eta} + \boldsymbol{\varepsilon},$$

In this context, we design the prior distributions as follows:

$$\begin{gathered}
\boldsymbol{\eta}|\boldsymbol{L}, \boldsymbol{w}, \sigma^2 \sim MVN(\boldsymbol{L}\boldsymbol{w}, \sigma^2 \boldsymbol{I}), \\
w_j|\sigma^2, \gamma_j = 0 \sim \mathcal{N}(0, \sigma^2 \kappa_0^2), \\
w_j|\sigma^2, \gamma_j = 1 \sim \mathcal{N}(0, \sigma^2 \kappa_1^2), \qquad (4.6) \\
\gamma_j \sim Bernoulli(q), P(\gamma_j = 1) = 1 - P(\gamma_j = 0) = q \\
\sigma^2 \sim IG(\alpha_1, \alpha_2)
\end{gathered}$$

where $0 < \kappa_0^2 < \kappa_1^2 < \infty$. Here, MVN stands for the multivariate normal distribution, $\boldsymbol{I}$ is the identity matrix, and $IG(\alpha_1, \alpha_2)$ is the inverse gamma distribution with shape α_1 and scale α_2. The spike-and-slab prior on the fusion coefficients $\boldsymbol{w}$ is a mixture of two distributions: named the "spike" and the "slab" distributions. An illustration of both distributions is shown in Figure 4.1, with the red dashed line representing the spike distribution and the blue dotted line representing the slab distribution. The x-axis represents the fusion coefficient values and the y-axis represents their probability density. The spike distribution follows a normal distribution that focuses most of its probability density around zero with very small variance κ_0^2, which encourages most of the fusion coefficients w_j to be uninformative. The slab distribution on the other hand, is a diffuse prior with large variance κ_1^2 to encourage exploration of different values for w_j. Then, the spike and slab prior (shown in black solid) is a mixture of the two distributions, allowing the model to obtain sparse solutions while sufficiently exploring the parameter space. Here, the latent binary indicators are used to denote which distribution w_j is sampled from. If $\gamma_j = 1$, then w_j is sampled from the slab distribution, while $\gamma_j = 0$ means that w_j is sampled from the spike distribution. For these latent binary indicators, we impose a Bernoulli prior on each γ_j such that $\gamma_j \sim Bernoulli(q)$. Note that q is a hyperparameter that is either pre-specified or sampled based on a hyperprior distribution. In particular, we follow existing recommendations [93] and set q such that $P\left(\sum_{j=1}^{s} \mathbb{I}(\gamma_j = 1) > \max(10, N)\right) = 0.1$with $\mathbb{I}$ as the indicator function. In general, this condition encourages the model to return sparse solutions by controlling the probability of the number of informative sensors. One can relax or strengthen the sparsity of the solutions by increasing/decreasing 10 in the $\max(10, N)$. Finally, for the variance noise σ^2, we impose an inverse Gamma prior. The inverse Gamma prior not only allows computationally efficient sampling by being a conjugate prior to the normal distribution, but also allows the proposed model

to have nice theoretical properties which will be covered in detail in Section 4.2.3. For the simulation and case studies, we set $(\alpha_1, \alpha_2) = (1,1)$ to let the inverse Gamma distribution act like an uninformative prior distribution. Additional experiments found that the values (α_1, α_2) have negligible effects on the parameter estimation.

After defining the prior distributions for the parameters, the next step is to derive the joint posterior distribution. Bayes' rule tells us that the joint posterior distribution is proportional to the prior distribution times the likelihood, i.e.,

$$P(\boldsymbol{w}, \sigma^2, \boldsymbol{\gamma}|\boldsymbol{\eta}) \propto P(\boldsymbol{\eta}|\boldsymbol{w}, \sigma^2, \boldsymbol{\gamma})P(\boldsymbol{w}, \sigma^2, \boldsymbol{\gamma}).$$

Here, the exact expression of the prior distribution $P(\boldsymbol{w}, \sigma^2, \boldsymbol{\gamma})$ is:

$$P(\boldsymbol{w}, \sigma^2, \boldsymbol{\gamma}) \propto P(\sigma^2)P(\boldsymbol{\gamma})P(\boldsymbol{w}|\boldsymbol{\gamma}, \sigma^2)$$

$$= P(\sigma^2)\left\{\prod_{j=1}^{s}\left[\left((1-q)\phi\left(w_j, 0, \kappa_0^2\sigma^2\right)\right)^{1-\gamma_j} + \left(q\phi\left(w_j, 0, \kappa_1^2\sigma^2\right)\right)^{\gamma_j}\right]\right\}.$$

where $\phi\left(w_j, 0, \kappa_0^2\sigma^2\right)$ is the probability density function of the normal distribution with mean 0 and variance $\kappa_0^2\sigma^2$ evaluated at w_j. Also, we follow existing works [94] and assume that the priors for $\boldsymbol{\gamma}$ and σ^2 are independent.

The likelihood distribution $P(\boldsymbol{\eta}|\boldsymbol{w}, \sigma^2, \boldsymbol{\gamma})$ follows a multivariate normal distribution with mean $\boldsymbol{Lw}$ and variance $\sigma^2\boldsymbol{I}$. Thus, the posterior distribution $P(\boldsymbol{w}, \sigma^2, \boldsymbol{\gamma}|\boldsymbol{\eta})$ does not have a closed form expression, so we have to resort to numerical methods for sampling. In particular, we use Gibbs sampling since the conditional distribution for each parameter has a closed form expression. Gibbs sampling, also known as alternating conditional sampling, is a computationally efficient approach to drawing samples from the posterior distribution [95]. Generally, given a parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ with d dimensions, the Gibbs sampler cycles through the subvectors of $\boldsymbol{\theta}$ at each

iteration t and draws each subset conditional on the values of all the others (i.e., $\theta_j \sim P(\theta_j|\boldsymbol{\theta}_{-j}^{t-1}, \mathcal{H})$), where $\boldsymbol{\theta}_{-j}^{t-1} = (\theta_1^t, \dots, \theta_{j-1}^t, \theta_{j+1}^{t-1}, \dots, \theta_d^{t-1})$, $j \in \{1, \dots, d\}$ and $\mathcal{H}$ represents some observed data.

The conditional distributions of all the parameters have closed forms due to the careful choice of prior distributions. The analytical expressions for the conditional distributions are listed below:

$$\boldsymbol{w}|\boldsymbol{\gamma}, \sigma^2, \boldsymbol{\eta}, \boldsymbol{L} \sim MVN(\boldsymbol{V}\boldsymbol{L}^T\boldsymbol{\eta}, \sigma^2\boldsymbol{V}),$$

$$\boldsymbol{V} = \left(\boldsymbol{L}^T\boldsymbol{L} + \boldsymbol{D}_\gamma\right)^{-1},\ \boldsymbol{D}_\gamma = \mathrm{Diag}(\boldsymbol{\gamma}\kappa_1^{-2} + (\boldsymbol{1} - \boldsymbol{\gamma})\kappa_0^{-2});$$

$$P\left(\gamma_j = 1 \middle| \boldsymbol{w}, \boldsymbol{\eta}, \sigma^2, \boldsymbol{L}\right) = \frac{q\phi(w_j, 0, \sigma^2\kappa_1^2)}{q\phi\left(w_j, 0, \sigma^2\kappa_1^2\right) + (1-q)\phi(w_j, 0, \sigma^2\kappa_0^2)};$$

$$P(\sigma^2|\boldsymbol{w}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{L}) \propto IG(\alpha_1', \alpha_2'),$$

$$\alpha_1' = \alpha_1 + \frac{\sum_{i=1}^N n_i}{2} + \frac{s}{2},$$

$$\alpha_2' = \alpha_2 + \frac{\boldsymbol{w}^T\boldsymbol{D}_\gamma\boldsymbol{w}}{2} + \frac{(\boldsymbol{\eta} - \boldsymbol{L}\boldsymbol{w})^T(\boldsymbol{\eta} - \boldsymbol{L}\boldsymbol{w})}{2}.$$

One challenge here is that the underlying degradation status $\boldsymbol{\eta}$ is unobservable, hence we cannot directly sample from the distributions above. To overcome this challenge, we utilize the definition on equation (4.1) that the underlying degradation status at the failure time is equal to the failure threshold (i.e., $\eta_i(T_i) = l$). Letting τ_i denote the observed failure time of unit i, we can approximate the degradation status $\eta_i(\tau_i) = l$ for all units i. Specifically, we can rewrite the conditional distributions above by replacing $\boldsymbol{\eta}$ with $\boldsymbol{\eta}(\boldsymbol{\tau}) = [\eta_1(\tau_1), \dots, \eta_N(\tau_N)]^T = l\boldsymbol{1}_N \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{L}$ with $\boldsymbol{L}(\boldsymbol{\tau}) = [\boldsymbol{L}_1(\tau_1), \dots, \boldsymbol{L}_N(\tau_N)]^T \in \mathbb{R}^{N\times s}$. In other words, we replace the top equation $\boldsymbol{\eta}|\boldsymbol{L}, \boldsymbol{w}, \sigma^2 \sim MVN(\boldsymbol{L}\boldsymbol{w}, \sigma^2\boldsymbol{I})$ in equation (4.6) with

$$l\boldsymbol{1}_N \approx \boldsymbol{\eta}(\boldsymbol{\tau})|\boldsymbol{L}(\boldsymbol{\tau}), \boldsymbol{w}, \sigma^2 \sim MVN(\boldsymbol{L}(\boldsymbol{\tau})\boldsymbol{w}, \sigma^2\boldsymbol{I}). \tag{4.7}$$

Our studies show that the failure threshold l acts as a scaling factor. Therefore, we can set l to any arbitrary positive number and finally normalize $\boldsymbol{w}$ if the value of l is not known. This will not affect our sensor selection nor prognostic results. The final conditional distributions used in the Gibbs sampler are illustrated below:

$$\boldsymbol{w}|\boldsymbol{\gamma}, \sigma^2, \boldsymbol{L}(\boldsymbol{\tau}) \sim MVN(\boldsymbol{V}\boldsymbol{L}(\boldsymbol{\tau})^T l\mathbf{1}_N, \sigma^2\boldsymbol{V}),$$

$$\boldsymbol{V} = \left(\boldsymbol{L}(\boldsymbol{\tau})^T\boldsymbol{L}(\boldsymbol{\tau}) + \boldsymbol{D}_\gamma\right)^{-1}, \boldsymbol{D}_\gamma = \mathrm{Diag}(\boldsymbol{\gamma}\kappa_1^{-2} + (\mathbf{1} - \boldsymbol{\gamma})\kappa_0^{-2});$$

$$P\left(\gamma_j = 1 \middle| \boldsymbol{w}, \boldsymbol{\eta}(\boldsymbol{\tau}), \sigma^2, \boldsymbol{L}(\boldsymbol{\tau})\right) = \frac{q\phi(w_j, 0, \sigma^2\kappa_1^2)}{q\phi\left(w_j, 0, \sigma^2\kappa_1^2\right) + (1-q)\phi(w_j, 0, \sigma^2\kappa_0^2)};$$

$$P\left(\sigma^2 \middle| \boldsymbol{w}, \boldsymbol{\eta}(\boldsymbol{\tau}), \boldsymbol{\gamma}, \boldsymbol{L}(\boldsymbol{\tau})\right) \propto IG(\alpha_1', \alpha_2'),$$

$$\alpha_1' = \alpha_1 + \frac{N}{2} + \frac{s}{2},$$

$$\alpha_2' = \alpha_2 + \frac{\boldsymbol{w}^T\boldsymbol{D}_\gamma\boldsymbol{w}}{2} + \frac{(l\mathbf{1}_N - \boldsymbol{L}(\boldsymbol{\tau})\boldsymbol{w})^T(l\mathbf{1}_N - \boldsymbol{L}(\boldsymbol{\tau})\boldsymbol{w})}{2}.$$

The main computational bottleneck of this approach is sampling from the multivariate normal distribution in $\boldsymbol{w}|\boldsymbol{\gamma}, \sigma^2, \boldsymbol{L}(\boldsymbol{\tau})$. The computational complexity of the proposed method is $\mathcal{O}(s^2 * (s \vee N))$, which can be demanding when the number of sensors s is extremely large. To mitigate this challenge, researchers have proposed workarounds such as using a block updating procedure [94] or a "skinny" Gibbs sampler [96], which reduces the complexity to $\mathcal{O}(N * (s \vee |\mathcal{A}|^2))$ albeit at the cost of running more Gibbs sampling iterations. Note that $|\mathcal{A}|$ represents the size of informative sensors. For our evaluations, we used the regular Gibbs sampling algorithm as it was sufficiently fast for offline training. Even in the most computationally demanding experimental setup with 1000 training units and 200 sensors, the Gibbs sampler on average took less than 5 minutes to run with 2000 warm-up iterations and 2000 sampling iterations.

Next, we discuss how to select the informative sensors based on the estimated joint posterior distribution. In particular, we determine that a sensor is informative when the marginal posterior probability $P(\gamma_j = 1|\boldsymbol{w}, \sigma^2) > 0.5$ such that:

$$\gamma_j = \begin{cases} 1, & if\ P\left(\gamma_j = 1\middle|\boldsymbol{w}, \boldsymbol{\eta}(\boldsymbol{\tau}), \sigma^2, \boldsymbol{L}(\boldsymbol{\tau})\right) \geq 0.5 \\ 0, & otherwise \end{cases}. \tag{4.8}$$

Barbieri et al. [97] showed that choosing 0.5 as the threshold value is predictively optimal (i.e., the model predictions achieve the minimum expected loss over the posterior distribution of the parameters). Hence, we also use the threshold of 0.5 in our context here.

4.2.3 Theoretical Properties

In this subsection, we investigate the important theoretical properties of the proposed Bayesian sensor selection approach. One important property during sensor selection is selection consistency, which investigates the conditions in which the selection algorithm can correctly select or fail to select the true informative sensors as the training sample size N increases. Please note that our discussion of selection consistency is concentrated on linear regression models with general spike and slab priors, as we will subsequently establish that the proposed model belongs to this class of models. Consider a conventional linear regression model $\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{e}$ with general spike and slab priors as such:

$$\begin{gathered} \boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2 \sim MVN(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I}), \\ \beta_j|\sigma^2, \gamma_j = 0 \sim \mathcal{D}_0 = \frac{1}{\kappa_0} f\left(\frac{x}{\kappa_0}\right), \\ \beta_j|\sigma^2, \gamma_j = 1 \sim \mathcal{D}_1 = \frac{1}{\kappa_1} f\left(\frac{x}{\kappa_1}\right), \\ P(\gamma_j = 1) = 1 - P(\gamma_j = 0) = q, \\ \sigma^2 \sim IG(\alpha_1, \alpha_2). \end{gathered} \tag{4.9}$$

Here, $\boldsymbol{Y} \in \mathbb{R}^{N\times 1}$ is the vector of response variables, $\boldsymbol{X} \in \mathbb{R}^{N\times s}$ is the design matrix, $\boldsymbol{\beta} \in \mathbb{R}^{s\times 1}$

are the regression coefficients, and $\boldsymbol{e} = [e_1, \dots, e_N] \in \mathbb{R}^{N\times 1}$ is the vector of independent Gaussian error terms such that $e_i \sim \mathcal{N}(0, \sigma^2)$ for $i \in \{1, \dots, N\}$. $\mathcal{D}_0$ and $\mathcal{D}_1$ denote the general spike and slab prior distributions for a base density $f(x)$, which is assumed to be unimodal, symmetric, and continuous. We observe that the proposed model is a special case of this general formulation by replacing $\boldsymbol{Y}$ with $l\mathbf{1}_N$, $\boldsymbol{\beta}$ with the fusion coefficients $\boldsymbol{w}$, $\boldsymbol{X}$ with $\boldsymbol{L}(\boldsymbol{\tau})$, $\boldsymbol{e}$ with $\boldsymbol{\varepsilon}(\boldsymbol{\tau}) = [\varepsilon_1(\tau_1), \dots, \varepsilon_N(\tau_N)]^T$ and $f(x)$ with $\frac{1}{\sqrt{2\pi}}\exp(-\frac{x^2}{2})$ in equation (4.6) and equation (4.7).

Under this general formulation, there are two versions of selection consistency: weak selection consistency (WSC) and strong selection consistency (SSC). WSC requires the posterior probabilities of the binary indicators γ_j to uniformly converge to the true values:

$$\min_{j=1,\dots,s} P\left(\gamma_j = t_j \middle| \boldsymbol{L}\right) \xrightarrow{\text{p}} 1. \tag{4.10}$$

Each t_j represents the true value of each sensor (i.e., whether the sensor is informative $t_j = 1$ or uninformative $t_j = 0$), and $\xrightarrow{\text{p}}$ denotes convergence in probability. On the other hand, SSC requires the posterior probabilities on the true model to converge to 1, such that:

$$P(\boldsymbol{\gamma} = \boldsymbol{t}|\boldsymbol{L}) \xrightarrow{\text{p}} 1, \tag{4.11}$$

where $\boldsymbol{t} = [t_1, \dots, t_s]$ indicates the set of ground truth values. The difference between the two types of selection consistency is that WSC focuses on the convergence of the individual γ_j to the true values t_j, but does not guarantee that the overall identified model is equal to the true model. Since SSC is a statement on the joint convergence of all $\boldsymbol{\gamma}$ to the true model $\boldsymbol{t}$, it is a more stringent condition than WSC.

Next, we discuss the conditions needed to achieve either WSC or SSC. In particular, the two conditions are listed below:

$$O_N \to 0; U_N \to \infty, \tag{4.12}$$

$$sO_N \to 0; \frac{U_N}{|\boldsymbol{t}|} \to \infty, \tag{4.13}$$

where the terms O_N and U_N are defined as such:

$$O_N := \sup_{|a| \le R_N, |b| \le R_N} \frac{q\mathcal{D}_1(a)}{(1-q)\mathcal{D}_0(b)},$$

$$U_N := \inf_{m_N \le |a|, m_N \le |b|, |a-b| \le \epsilon_N} \frac{q\mathcal{D}_1(a)}{(1-q)\mathcal{D}_0(b)}.$$

Here, we introduce the minimal signal strength m_N, which is defined as:

$$\min_j |\beta_j| \ge m_N := \sqrt{\frac{C\sigma^2 \log s}{N}},$$

for some large enough constant $C > 0$. Intuitively, the minimal signal strength m_N ensures that the signals from the truly informative sensors are strong enough to be distinguished from noise. If the true coefficients β_j are not sufficiently large (i.e., smaller than m_N), then it is difficult to identify the informative sensors from the pool of sensors. Finally, ϵ_N ensures that the point in the slab prior a and the point in the spike prior b is located reasonably close to each other such that $\frac{1}{\sqrt{N}} \le \epsilon_N \to 0$. The quantity O_N represents the magnitude of the slab prior relative to the spike prior in the neighborhood of the origin with radius R_N, while the quantity U_N indicates the same relative magnitude but instead around the distribution tails. Ideally, we want O_N going to zero and U_N going to infinity as N increases (i.e., concentrated spike mass near the origin to induce sparse solutions, while flatter tails of the slab prior to encourage exploration).

Based on existing works [87], the 1st condition in equation (4.12) guarantees WSC for the entire general spike and slab prior setting in equation (4.9), which includes the proposed model. In addition, if the 2nd condition in equation (4.13) holds, we have SSC. It can be seen that SSC

requires a stronger condition since O_N needs to go to zero more quickly and U_N needs to go to infinity more quickly.

To satisfy both conditions and achieve SSC, [93] suggested to set the prior variances κ_0^2 and κ_1^2 depend on the sample size N. Therefore, inspired by [93], we impose the following sufficient conditions to satisfy SSC:

$$\kappa_0^2 \to 0, N\kappa_1^2 \approx (N + s^{2+\omega}), \tag{4.14}$$

Here, the notation $\approx$ denotes that the two quantities on both sides of the equation have the same order, and $\omega > 0$ denotes an arbitrary fixed positive number. To see how these conditions in equation (4.14) assist in satisfying conditions in equation (4.12) and equation (4.13), without loss of generalization, we first define analogous quantities of O_N and U_N based on a radius of $R_N = \sqrt{\frac{(2+\omega)\log s}{N}}$, while setting $\sigma^2 = 1$:

$$O_N' = \frac{q\kappa_0 f(0)}{(1-q)\kappa_1 f\left(\frac{R_N}{\kappa_0}\right)},$$

$$U_N' = \frac{q\kappa_0 f(\frac{m_N}{\kappa_1})}{(1-q)\kappa_1\left(f\left(\frac{m_N}{2\kappa_0}\right) + \kappa_0 \exp\left(-\frac{Nm_N^2}{2}\right)\right)}.$$

Based on the definition of the minimum signal strength m_N above, we can deduce that R_N is a slightly larger radius than the minimum signal strength m_N when we set $C = 2$. Here, we can see that O_N' is equivalent to O_N since it represents the same ratio of the spike distribution and the slab distribution at the origin with radius R_N. Similarly, U_N' is equivalent to U_N since it denotes the same ratio of the spike distribution and the slab distribution near the tails (i.e., outside the minimal signal strength m_N).

Next, to satisfy SSC, we need to verify that the analogous quantities O'_N and U'_N satisfy equation (4.13), i.e.,

$$sO'_N \to 0; \frac{U'_N}{|\boldsymbol{t}|} \to \infty.$$

Since the first condition in equation (4.14) states that $\kappa_0^2 \to 0$, we start by plugging in $\kappa_0 = \frac{1}{\sqrt{N}}$ and analyzing the behavior of κ_1. By setting $f(x) = \exp(-\frac{x^2}{2})$, we have the following results for O'_N and U'_N:

$$O'_N = \frac{q}{(1-q)\sqrt{N}\phi\left(\sqrt{-\frac{(2+\omega)\log s}{2}}\right)} \times \frac{1}{\kappa_1} = \frac{q}{(1-q)} \times \frac{s^{\zeta}}{(N\kappa_1^2)^{\frac{1}{2}}} \tag{4.15}$$

$$U'_N = \frac{\frac{q}{\sqrt{N}}}{(1-q)} \times \frac{s^{-\frac{\lambda}{N\kappa_1^2}}}{\kappa_1} \times \frac{1}{s^{-\frac{\lambda}{4}} + s^{-\lambda}} \geq \frac{q}{(1-q)} \times \frac{s^{\lambda\left(1-\frac{1}{N\kappa_1^2}\right)}}{(N\kappa_1^2)^{\frac{1}{2}}} \tag{4.16}$$

where ζ is a positive constant such that $\zeta = \frac{2+\omega}{2}$, and λ is a constant that depends on the minimal signal strength via

$$\lambda = \frac{Nm_N^2}{2\log s}.$$

Then, imposing the second condition $N\kappa_1^2 \approx (N + s^{2+\omega})$ on equations (4.15) and (4.16) unveils two key observations. The first observation is that the $N\kappa_1^2$ term in the denominator of equation (4.15) will drive $sO'_N \to 0$, achieving WSC. The second observation is that the exponential term s^{λ} in equation (4.16) will dominate the numerator as $N \to \infty$, resulting in $U'_N \to \infty$. Since $|\boldsymbol{t}| \leq s$, $\frac{U'_N}{|\boldsymbol{t}|} \to \infty$ as well. In this way, O'_N and U'_N satisfy equation (4.13) and thus we achieve SSC. In the numerical evaluations, we use the following values for the prior variances.

$$\kappa_0^2 = \frac{1}{10N}, \kappa_1^2 = \max\left(\frac{s^{2.1}}{N}, \log N\right) \tag{4.17}$$

Note that the settings in equation (4.17) satisfy the properties in equation (4.14). This means that letting the prior variances for the spike and slab prior (i.e., κ_0^2 and κ_1^2) depend on N is a key component to achieve SSC.

4.2.4 Remaining Useful Life Prediction

Once the Gibbs sampler returns the estimated fusion coefficients $\widehat{\boldsymbol{w}}$ and the binary sensor indicators $\widehat{\boldsymbol{\gamma}}$, we can easily construct the HI for a given historical unit i. Recall that the uninformative sensor's indicators will be zeroed out based on the sensor selection procedure described in Section 4.2.3. Thus, unit i's HI is defined as $\boldsymbol{h}_i = \boldsymbol{L}_i \text{Diag}(\widehat{\boldsymbol{\gamma}})\widehat{\boldsymbol{w}}$. The same procedure can be done for an in-service unit r, where its HI is represented as $\boldsymbol{h}_r = \boldsymbol{L}_r \text{Diag}(\widehat{\boldsymbol{\gamma}})\widehat{\boldsymbol{w}}$. Since we do not make any assumptions on the underlying degradation process $\eta(t)$, we can estimate the RUL of the in-service unit r using a wide range of degradation models. Here, we use a general path model to estimate the RUL such that $h_i(t) = \eta_i(t) + \varepsilon_i(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_i + \varepsilon_i(t)$, where $\boldsymbol{\psi}(t) = [1, t, \dots, t^{M-1}]$ is the $(M-1)$-order polynomial basis function and $\boldsymbol{\Gamma}_i = \left[\Gamma_{i,1}, \dots, \Gamma_{i,M}\right]^T \in \mathbb{R}^{M\times 1}$ are the corresponding random-effect coefficients. Note that to increase the model flexibility, we can also use any generic basis functions, such as B-splines depending on the model fit.

The random effect parameter is assumed to follow a prior distribution such that $\boldsymbol{\Gamma}_i \sim G(\cdot)$, where $G(\cdot)$ is typically estimated from the historical units. For the in-service unit r, we can estimate the posterior distribution of $\boldsymbol{\Gamma}_r$ via the Bayes' rule such that $P(\boldsymbol{\Gamma}_r|\boldsymbol{h}_r) \propto P(\boldsymbol{\Gamma}_r)P(\boldsymbol{h}_r|\boldsymbol{\Gamma}_r)$. If the posterior distribution does not have an analytical solution, we can use numerical methods like Hamiltonian Monte Carlo [98] to sample from the posterior distribution. The cumulative distribution function (CDF) of the failure time T_r of the in-service unit r can then be expressed as

$F_{T_r}(t|\boldsymbol{h}_r) = P(T_r \le t|\boldsymbol{h}_r) = P(\boldsymbol{\psi}(t)\boldsymbol{\Gamma}_r \ge l|\boldsymbol{h}_r)$ based on the definition in equation (4.1).

Given that the in-service unit r has not yet failed, the failure time CDF can be further updated using the last observed measurement time t_{r,n_r} as such:

$$F_{T_r}\left(t\middle|\boldsymbol{h}_r, T_r > t_{r,n_r}\right) = \frac{P(\boldsymbol{\psi}(t)\boldsymbol{\Gamma}_r \ge l|\boldsymbol{h}_r) - P(\boldsymbol{\psi}\left(t_{r,n_r}\right)\boldsymbol{\Gamma}_r \ge l|\boldsymbol{h}_r)}{1 - P(\boldsymbol{\psi}\left(t_{r,n_r}\right)\boldsymbol{\Gamma}_r \ge l|\boldsymbol{h}_r)}.$$

If we assume that $G(\cdot)$ follows a multivariate normal distribution (i.e., $\boldsymbol{\Gamma}_r \sim MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$), then the posterior distribution $\boldsymbol{\Gamma}_r|\boldsymbol{h}_r$ also follows a multivariate normal distribution with a closed form expression such that $\boldsymbol{\Gamma}_r|\boldsymbol{h}_r \sim MVN(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$, where

$$\boldsymbol{\Sigma}_r = \left(\frac{\boldsymbol{\Psi}_r^T\boldsymbol{\Psi}_r}{\sigma^2} + \boldsymbol{\Sigma}^{-1}\right)^{-1}, \boldsymbol{\mu}_r = (\boldsymbol{\Sigma}_r)^{-1}\left(\frac{\boldsymbol{\Psi}_r^T\boldsymbol{h}_r}{\sigma^2} + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right).$$

The conditional CDF can then be rewritten as such:

$$F_{T_r}\left(t\middle|\boldsymbol{h}_r, T_r > t_{r,n_r}\right) = \frac{\Phi\big(g(t)\big) - \Phi\left(g\left(t_{r,n_r}\right)\right)}{1 - \Phi\left(g\left(t_{r,n_r}\right)\right)}. \tag{4.18}$$

where $\Phi(\cdot)$ is the standard normal distribution CDF, and $g(t) = (\boldsymbol{\psi}(t)\boldsymbol{\mu}_r - l)/(\boldsymbol{\psi}(t)\boldsymbol{\Sigma}_r\boldsymbol{\psi}(t)^T)^{0.5}$. Following existing studies [4], we can use the median of $F_{T_r}\left(t\middle|\boldsymbol{h}_r, T_r > t_{r,n_r}\right)$ (i.e., $F_{T_r}\left(\hat{T}_r\middle|\boldsymbol{h}_r, T_r > t_{r,n_r}\right) = 0.5$) to account for the skewness in the truncated CDF. Hence, the estimated RUL is $\hat{T}_r - t_{r,n_r}$.

A comprehensive summary of the proposed method, including the prior specification, parameter estimation, sensor selection, and RUL prediction is provided in the flowchart in Figure 4.2.

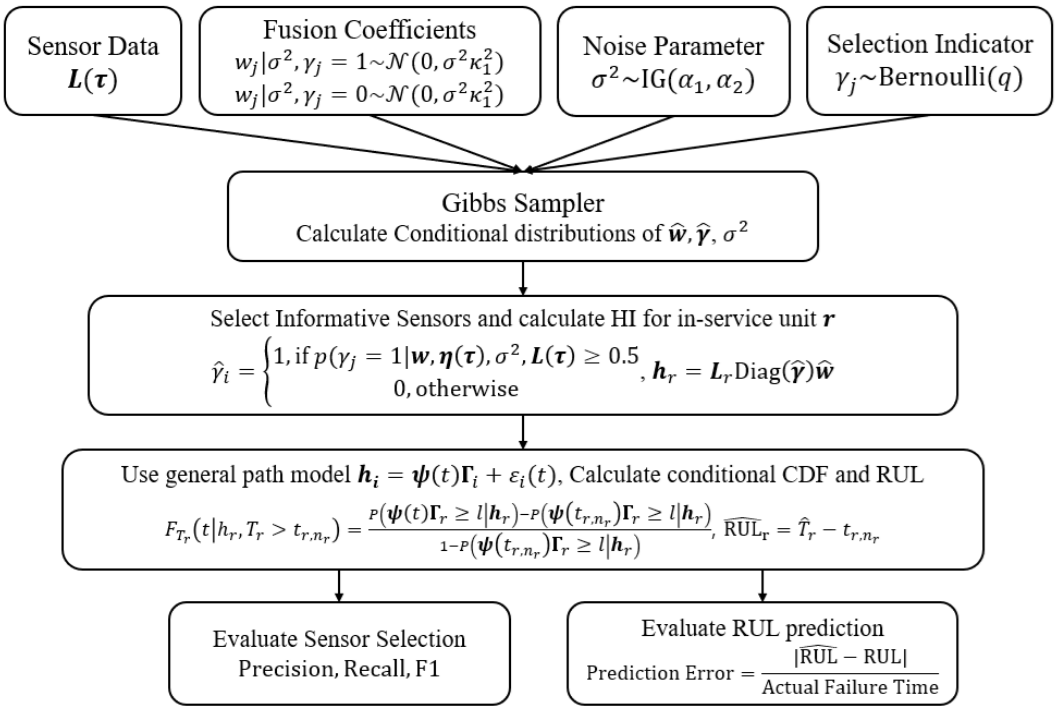


Figure 4.2 Flowchart of the Proposed Model

4.3 Simulation Studies

In this section, we conduct a series of simulation studies to evaluate the performance of the proposed method. Section 4.3.1 first discusses how we generate the sensor signals. Then, Section 4.3.2 introduces the benchmark methods and evaluation metrics. Then, we investigate the sensor selection performance under two different scenarios. Section 4.3.3 investigates the proposed model's sensor selection capabilities under varying number of sensors and training units. Section 4.3.4 then repeats the same study while considering the effect of various types of correlation.

4.3.1 Data Generation Settings

For the simulation study, the underlying degradation path $\eta_i(t)$ is defined via a linear degradation process such that $\eta_i(t) = \Gamma_{i,0} + \Gamma_{i,1}t$ and the random-effect parameter follows a multivariate normal distribution (i.e., $\boldsymbol{\Gamma}_i \sim MVN\left(\begin{pmatrix}-1\\3\end{pmatrix}, \begin{pmatrix}100 & 1\\1 & 0.5\end{pmatrix}\right)$). If the sampled $\Gamma_{i,1} \leq 0$ due to the multivariate normal distribution, then we discard the sample and sample a new one to guarantee monotonicity. In addition, the true failure threshold is set to $l = 100$. The true observed failure times of unit i are recorded using the definition in equation (4.1), while the true HI is created by adding a random Gaussian noise $h_i(t) = \eta_i(t) + \varepsilon_i(t),\ \varepsilon_i(t) \sim \mathcal{N}(0,10^2)$.

The units have both informative and uninformative sensors. Specifically, there are three informative sensors, while the remaining sensors are uninformative. Naturally, only the first three true fusion coefficients are nonzero, while the remaining true fusion coefficients are set to zero (i.e., $\boldsymbol{w} = [1.5, 2.0, 2.5, 0, \dots, 0] \in \mathbb{R}^{s\times 1}$). The three informative sensors are generated as:

$$\begin{aligned} L_{i,1}(t) &= \delta_{i,1}^{(1)}\sqrt{t} - \delta_{i,1}^{(2)}\sin(0.05t) + \varepsilon_{i,1}(t), \\ L_{i,2}(t) &= \delta_{i,2}^{(1)}t - \delta_{i,2}^{(2)}\sin(0.1t) + \varepsilon_{i,2}(t), \\ L_{i,3}(t) &= \frac{h_i(t) - w_1 L_{i,1}(t) - w_2 L_{i,2}(t)}{w_3}, \end{aligned} \tag{4.19}$$

where $\delta_{i,1}^{(1)}, \delta_{i,1}^{(2)}, \delta_{i,2}^{(1)} \sim \text{Uniform}(10,20), \delta_{i,2}^{(2)} \sim \text{Uniform}(0,2)$, and $\varepsilon_{i,1}(t), \varepsilon_{i,2}(t) \sim \mathcal{N}(0,10^2)$. The uninformative sensors are then generated as:

$$\begin{aligned} L_{i,u_1}(t) &= \left[\sum_{j=1}^{n_i} \delta_{i,u_1}^{(1)}(t_{i,j})\right] + \delta_{i,u_1}^{(2)} + \varepsilon_{i,u_1}(t), \\ L_{i,u_2}(t) &= \delta_{i,u_2}^{(1)}t + \delta_{i,u_2}^{(2)} + \varepsilon_{i,u_2}(t), \end{aligned} \tag{4.20}$$

where $\delta_{i,u_1}^{(1)} \sim \mathcal{N}(0,0.5^2), \delta_{i,u_1}^{(2)} \sim \text{Uniform}(10,30),\ \ \delta_{i,u_2}^{(1)} \sim \text{Uniform}(0,30)$, $\delta_{i,u_2}^{(2)} \sim \text{Uniform}(0,2)$, and $\varepsilon_{i,u_1}(t), \varepsilon_{i,u_2}(t) \sim \mathcal{N}(0,10^2)$.

Notice that the set of informative sensors are denoted by $\mathcal{I}$. Among the set of uninformative sensors $\mathcal{U}$, there are two types of uninformative sensors, each denoted by $\mathcal{U}_1$ and $\mathcal{U}_2$. Uninformative sensors belonging to $\mathcal{U}_1$ represent "random" sensors that act as noise since it is a stochastic process made up by summing random Gaussian terms. On the other hand, uninformative sensors belonging to $\mathcal{U}_2$ represent "consistent" sensors. These sensors have consistent increasing trends due to the linear relationship with time but are not related to the underlying degradation process. Note that these sensors can be mislabeled as informative sensors based on the heuristic approach [4], [74], [75] due to their increasing trends. For all simulations, we generate an approximately equal number of uninformative sensors from each category. For instance, if we set the total number of sensors as $s = 15$, then we will have 3 informative sensors, and $\frac{15-3}{2} = 6$ sensors of each "random" and "consistent" uninformative sensors. In addition, we assume that all signals are recorded at uniformly spaced time intervals. Figure 4.3 shows the trajectory of the 3 informative sensors and the true HI, while Figure 4.4 shows the trajectories of two types of uninformative sensors from a randomly selected unit.

4.3.2 Benchmark Methods

In this subsection, we review the benchmark methods and the evaluation metrics used in the simulation studies. In particular, we evaluate the proposed model's sensor selection performance against other popular sensor selection algorithms. The first class of competing models are likelihood penalization methods, which attach a penalty term to the likelihood function to promote sparse solutions. Specifically, we consider the adaptive LASSO [83], SCAD [84], and MCP [99] methods. Adaptive LASSO uses the L1 norm of the fusion coefficients as a penalty, in which larger penalty weights are imposed on less important sensors. SCAD imposes the same penalty as the adaptive LASSO for small fusion coefficients but imposes a relatively more relaxed constant

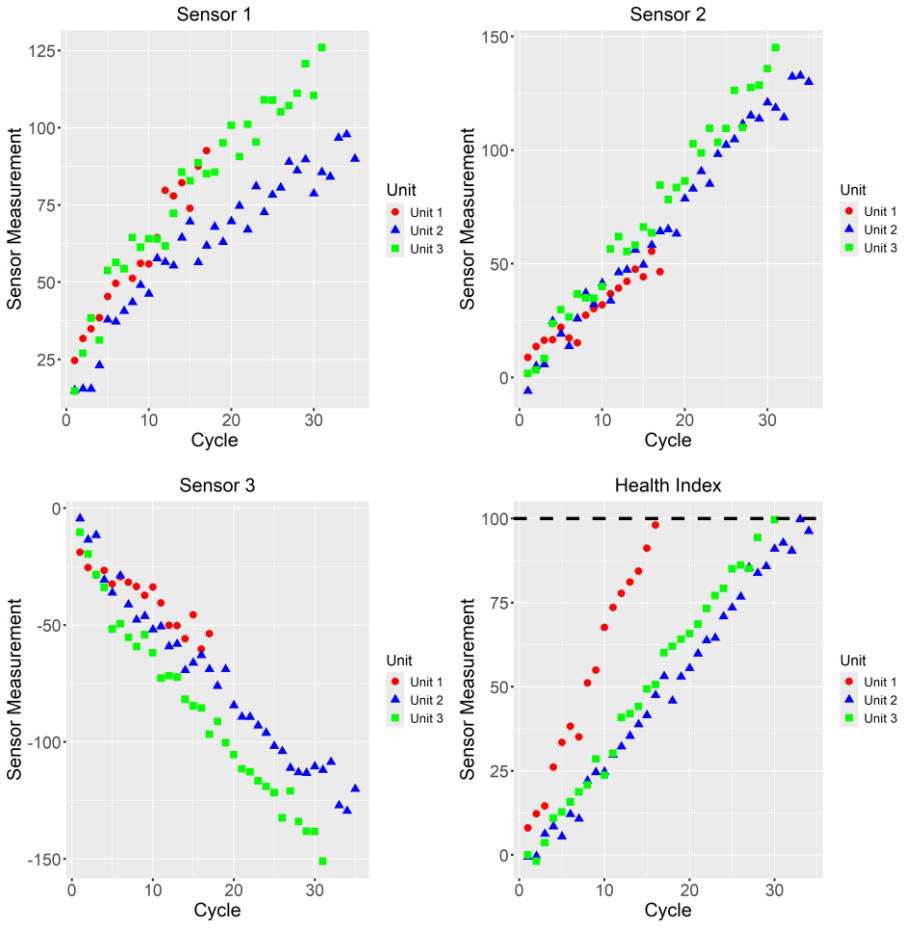


Figure 4.3 Plot of the Three Informative Sensors and the True HI of Three Randomly Generated Units

penalization rate for larger fusion coefficients, resulting in lower bias in the fusion coefficient estimation. MCP is also similar to SCAD, but it relaxes the penalization rate more quickly for larger fusion coefficients. Furthermore, we also include a different Bayesian variable selection approach named variational Bayes (VB). The VB approach showed promising variable selection performance in high dimensions based on a case study using genomic data [100].

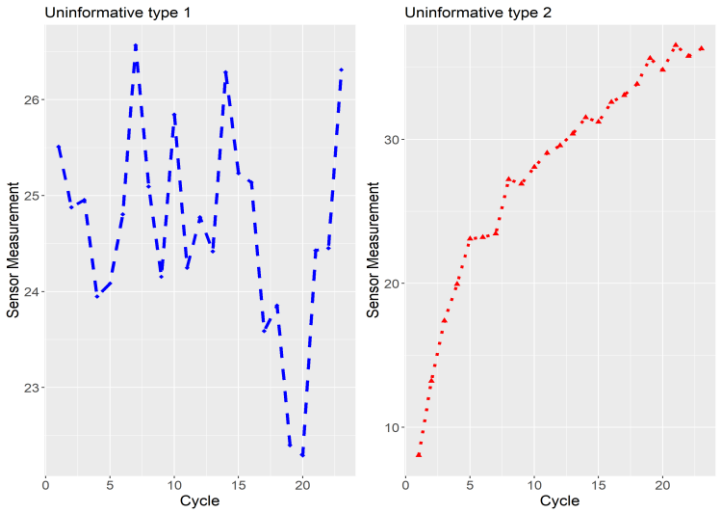


Figure 4.4 Plot of the Two Types of Uninformative Sensors in a Randomly Generated Unit

For the proposed method, we need to specify the failure threshold l to estimate the fusion coefficients $\boldsymbol{w}$. According to Section 4.2.2, $\boldsymbol{w}|\boldsymbol{\gamma}, \sigma^2, \boldsymbol{L}(\boldsymbol{\tau}) \sim MVN(\boldsymbol{V}\boldsymbol{L}(\boldsymbol{\tau})^T l\mathbf{1}_N, \sigma^2\boldsymbol{V})$. Thus, the failure threshold l only acts as a scaling factor for the mean of the Gibbs updating distribution for $\boldsymbol{w}$ and does not affect RUL prediction. Here, we simply use $l = 100$ when estimating the fusion coefficients for simplicity.

We use the following settings for all of the simulation studies. For the adaptive LASSO, we use the "glmnet" library [101] in R with 5-fold cross validation to find the optimal shrinkage parameter. In addition, we follow the recommendations of [4] and use $1/|\boldsymbol{w}_{OLS}|$ as the penalty weights for the fusion coefficients. Note that $\boldsymbol{w}_{OLS}$ is the ordinary least squares (OLS) estimate of $\boldsymbol{w}$. For the SCAD and MCP model, we use the "ncvreg" library [99] in R with max iterations set to 3000. For

the VB model, we use the "varbvs" library [100] with a Gaussian family. For the proposed method, we use a Gibbs sampler with 2000 warm-up iterations and 2000 sampling iterations. Similar to existing works [74], we use the polynomial basis functions $\boldsymbol{\psi}(t) = [1, t, t^2]$ for all methods.

We assess the model's ability to maximize true positives (i.e., selecting the informative sensors) while minimizing false positives (i.e., selecting the uninformative sensors). Note that minimizing false positives corresponds to reducing sensor misclassification costs. Hence, we consider the informative sensors as positive labels and uninformative sensors as negative labels and apply widely used classification metrics for selection performance evaluation. In particular, we consider three metrics: precision, recall, and F1. Precision measures the proportion of sensors identified by the model as informative that are indeed informative, while recall measures the proportion of actual informative sensors that were retrieved by the model. F1 balances out both precision and recall by taking the harmonic mean. Detailed definitions are listed below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

where TP stands for true positive, FP is false positive, and FN is false negative. With these metrics, we aim to provide a holistic evaluation of the competing methods.

4.3.3 Sensor Selection Performance without Correlation

In this subsection, we investigate the sensor selection performance of the proposed model with varying number of sensors and training units without the effect of correlation. Note that there are two types of correlation: intra-correlation, and inter-correlation. Intra-correlation refers to the correlation within each group of informative and uninformative sensors, while inter-correlation refers to the correlation between the groups of informative and uninformative sensors. We will discuss how to impose each type of correlation and investigate its effect on sensor selection in the

following Section 4.3.4. Here, we consider neither type of correlation.

Recall that the two main factors that characterize the high-dimensional scenarios are: 1) a low number of training units N; and 2) a high number of sensors s. Therefore, we conduct two different simulations that reflect these two conditions. First, we fix the number of sensors at $s = 45$ (i.e., 3 informative and 42 uninformative) and vary the number of training units N from 25 to 100. Second, we fix the number of training units to 50 and vary the number of sensors s from 15 to 75. Note that the number of informative sensors is fixed at 3 for all different configurations of s. Generally, we expect that a low N and a high s scenario will be the most challenging for sensor selection. Results of the evaluation are shown in Figure 4.5. Note that we display the mean results obtained from 100 repeated iterations.

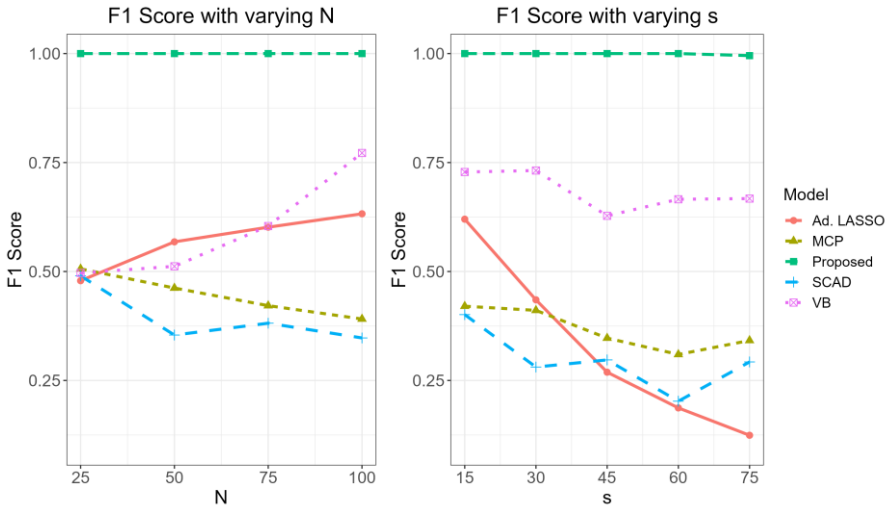


Figure 4.5 Sensor Selection Performance Regarding F1 Score with respect to Varying N and s.

From the results in Figure 4.5, the proposed method drastically outperforms existing methods in terms of the F1 score across all scenarios. In particular, the proposed method shows near to perfect performance even in challenging circumstances with small N and large s. However, likelihood penalization methods like adaptive LASSO, MCP, and SCAD suffer from many false

positives and generally perform worse in terms of F1 score as N decreases and s increases. The VB method generally outperforms other penalized likelihood methods, but still falls short relative

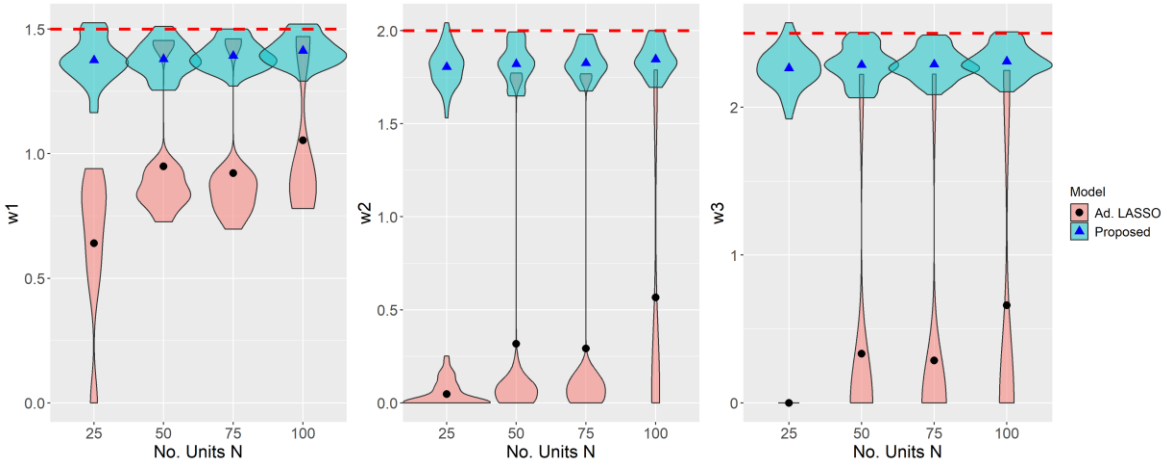


Figure 4.6 Plot of the Estimated Fusion Coefficients Between the Proposed Method (Blue) and Adaptive LASSO (pink) with varying N.

to the proposed method.

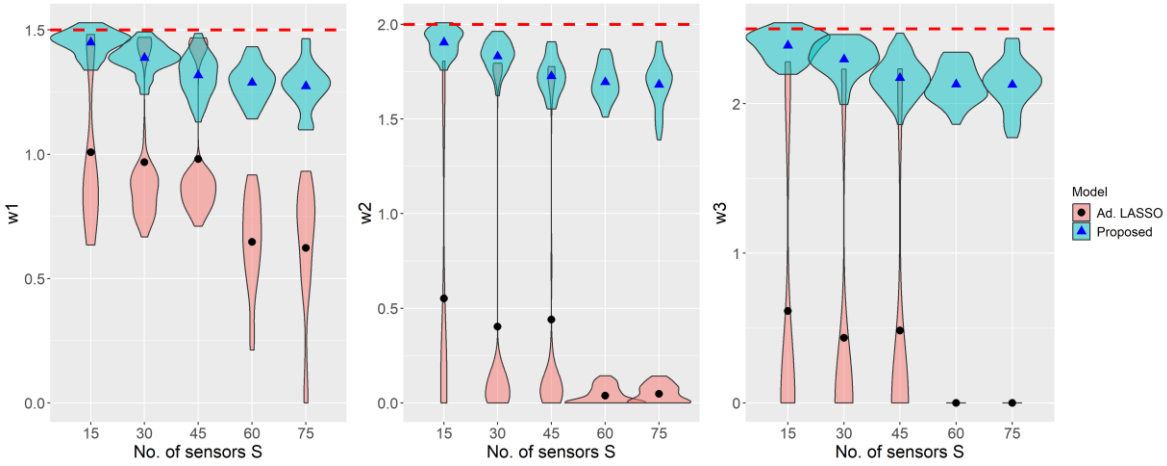


Figure 4.7 Plot of the Estimated Fusion Coefficients Between the Proposed Method (Blue) and Adaptive LASSO (pink) with varying s.

In addition to the sensor selection results, we examine the fusion coefficient estimation results as well under the same simulation settings. In particular, we repeat the fusion coefficient estimation 100 times and compare the estimates with the fusion coefficients estimated by adaptive LASSO. Kim et al. [4] have shown that adaptive LASSO is a simple yet one of the best sensor selection

approaches for HI construction, so we use it as a benchmark. Figure 4.6 shows the estimation results with varying N from 25 to 100 (fixed $s = 45$) , while Figure 4.7 shows the estimation results with varying s from 15 to 75 (fixed $N = 50$). The true coefficient values are shown in the red dashed line on the top, while the proposed model's fusion coefficients are shown in blue violin plots and the adaptive LASSO's coefficients are shown in pink violin plots. The average coefficient estimates of each setting are marked by the blue triangle (proposed) and black dots (adaptive LASSO). From the figures, we observe that the proposed model has drastically more accurate fusion coefficient estimates than the adaptive LASSO. Even in very high-dimensional settings (i.e., $N = 25$ in Figure 4.6 or $s = 75$ in Figure 4.7), the proposed method is able to obtain accurate and stable fusion coefficient estimates. On the contrary, the adaptive LASSO's coefficients are both inaccurate and unstable (i.e., high variance in the violin plots).

We also conducted additional experiments by fixing the number of sensors to $s = 50$ and varying the number of informative sensors from 3 to 30. The results showed that the model performs best when the true model is sparse, with much fewer informative sensors than the uninformative sensors.

4.3.4 Sensor Selection Performance with Correlation

In this subsection, we further investigate the sensor selection performance with correlation. Specifically, we introduce intra-correlation within each group of informative and uninformative sensors via correlated errors. The correlation between the informative sensors is imposed as

$$\boldsymbol{\Sigma}_{\mathcal{I}} = Cor\left(\varepsilon_{i,1}(t), \varepsilon_{i,2}(t)\right) = \begin{pmatrix} 1 & \rho_{inf} \\ \rho_{inf} & 1 \end{pmatrix} \in \mathbb{R}^{2\times 2},$$

where $\boldsymbol{\Sigma}_{\mathcal{I}}$ represents the corresponding correlation matrix, and $-1 < \rho_{inf} < 1$ controls the level of correlation. According to equation (4.19), all $L_{i,1}(t), L_{i,2}(t), L_{i,3}(t)$ are now correlated since

$L_{i,1}(t)$ and $L_{i,2}(t)$ have correlated errors, and $L_{i,3}(t)$ is a function of the remaining two informative sensors. Similarly, the intra-correlation between the uninformative sensors is imposed as:

$$\boldsymbol{\Sigma}_{\mathcal{U}} = Cor\left(\varepsilon_{i,4}(t), \varepsilon_{i,5}(t), \dots, \varepsilon_{i,s}(t)\right) == \begin{pmatrix} 1 & \cdots & \rho_{uninf} \\ \vdots & \ddots & \vdots \\ \rho_{uninf} & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{(s-3)\times(s-3)},$$

where $-1 < \rho_{uninf} < 1$ controls the level of correlation, and $\left\{\varepsilon_{i,4}(t), \varepsilon_{i,5}(t), \dots, \varepsilon_{i,\lfloor\frac{s-3}{2}\rfloor}(t)\right\} \in \mathcal{U}_1$ and $\left\{\varepsilon_{i,\lfloor\frac{s-3}{2}\rfloor+1}(t), \dots, \varepsilon_{i,s}(t)\right\} \in \mathcal{U}_2$. Next, we also consider inter-correlation between informative and uninformative sensors. This is expected to be the most damaging type of correlation as it can significantly interfere with the sensor selection procedure. To simulate this inter-correlation, we adopt a block-covariance setting as such:

$$\boldsymbol{\Sigma}_{total} = \begin{pmatrix} \boldsymbol{\Sigma}_{\mathcal{I}} & \boldsymbol{\Sigma}_{\mathcal{IU}}^{T} \\ \boldsymbol{\Sigma}_{\mathcal{IU}} & \boldsymbol{\Sigma}_{\mathcal{U}} \end{pmatrix} \in \mathbb{R}^{(s-1)\times(s-1)}, \boldsymbol{\Sigma}_{\mathcal{IU}} = \rho_{inter}\mathbf{1}_{(s-3)\times 2},$$

in which $-1 < \rho_{inter} < 1$ controls the level of inter-correlation. Finally, we use this block covariance matrix to sample the correlated errors as such:

$$\left[\varepsilon_{i,1}(t), \varepsilon_{i,2}(t), \varepsilon_{i,4}(t), \dots, \varepsilon_{i,s}(t)\right] \sim MVN(\mathbf{0}, \boldsymbol{\Sigma}_{total}).$$

Under this simulation setting, we demonstrate the effectiveness of the proposed method under weak and strong levels of correlation. In the first setting, we impose weak intra-correlation and inter-correlation of 0.25 (i.e., $\rho_{inf} = \rho_{uninf} = \rho_{inter} = 0.25$). In the next setting, we impose a much stronger level of intra and inter-correlation by setting $\rho_{inf} = 0.25, \rho_{uninf} = 0.90, \rho_{inter} = 0.75$. For the number of sensors and training units, we fix them to $s = 45$, $N = 50$ for both scenarios. The simulation results averaged across 100 iterations are shown in Table 4.2 and Table 4.1.

Table 4.2 Simulation Results Under Weak Correlation (standard deviations shown in parenthesis, $\rho_{inf} = \rho_{uninf} = \rho_{inter} = 0.25$)

Models	Precision	Recall	F1
Adaptive LASSO	0.446 (0.215)	0.367 (0.101)	0.381 (0.089)
MCP	0.751 (0.259)	0.500 (0.168)	0.548 (0.133)
SCAD	0.826 (0.294)	0.373 (0.109)	0.475 (0.098)
VB	0.970 (1.120)	0.400 (0.135)	0.564 (0.127)
Proposed	**0.995 (0.035)**	**1.000 (0.000)**	**0.997 (0.020)**

Results in Table 4.2 and Table 4.1 demonstrate the superior sensor selection performance of the proposed model. Indeed, except for the precision score in the strong correlation scenario, the proposed method drastically outperforms competing methods. Other competing methods noticeably produce numerous false positives in the presence of strong intercorrelation, with the adaptive LASSO even reaching an average F1 score below 0.4. The variational inference method excels at minimizing false positives with a higher precision score but produces too many false negatives, leading to a poor average F1 score of 0.555. On the other hand, the proposed method achieves a good balance between reducing false positives and false negatives. As a result, the

Table 4.1 Simulation Results Under Strong Correlation (standard deviations shown in parenthesis, $\rho_{inf} = 0.25, \rho_{uninf} = 0.90, \rho_{inter} = 0.75$)

Models	Precision	Recall	F1
Adaptive LASSO	0.389 (0.131)	0.371 (0.107)	0.370 (0.088)
MCP	0.591 (0.222)	0.629 (0.107)	0.573 (0.109)
SCAD	0.701 (0.302)	0.395 (0.131)	0.464 (0.103)
VB	**0.981 (0.092)**	0.400 (0.134)	0.555 (0.121)
Proposed	0.945 (0.120)	**1.000 (0.000)**	**0.967 (0.072)**

proposed method achieves a much higher average F1 score of 0.967 than any other competing methods.

4.4 Case Studies

In this section, we further evaluate the proposed method on a dataset of aircraft gas turbine engines. The results are also compared with the state-of-the-art benchmark approach: a generic HI model by Kim et al. [4] with 4 different approaches for sensor selection: Adaptive LASSO, MCP, SCAD, and VB. In particular, we first introduce the dataset in Section 4.4.1. Then, in Section 4.4.2, we compare the RUL prediction performance of the proposed model with the benchmarks. In Section 4.4.3, we consider a high-dimensional scenario by randomly reducing the number of training units. Finally, Section 4.4.4 further evaluates a high-dimensional scenario by augmenting the dataset with additional uninformative sensors.

4.4.1 Dataset Description

We use the turbofan engine dataset generated by C-MAPSS, a widely used simulation software developed by NASA. C-MAPSS has been widely used for studying the degradation process of large commercial turbofan engines [102]. The C-MAPSS dataset contains a total of four different sub-datasets with different failure modes and operating conditions. Here, we focus on the 1st sub-dataset (i.e., FD001) as it has a single failure mode with respect to the High-Pressure Compressor (HPC) and one operating condition. In addition, the simulated engines start with varying degrees of manufacturing variation and initial wear and tear to better mimic real-life degradation scenarios.

Each unit i in the dataset consists of 21 condition monitoring sensor signals measured at each cycle time $t = 1,2, \dots, n_i$. Details of the 21 sensors are provided in Table 4.3. The training set contains 20631 observations from 100 historical units, while the test set contains 13096

observations from 100 in-service units. Note that historical units in the training set contain measurements from start to failure, while the in-service units in the test set contain measurements from start-up to a random truncation time point prior to failure. The true RUL labels are provided

Table 4.3 C-MAPSS Sensor Information

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	rpm
Nc	Physical core speed rpm	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 correct4ed fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

for both the historical units and the in-service units. In addition, all sensors are first log-transformed, and then standardized as the existing literature (e.g., [91]).

Previous works eliminated sensors that did not exhibit a consistent monotonic trend or if their variance is less than 10^{-4} [4], [74]. However, the simulation results in Section 4.3 showed that the monotonic assumption is not a clear indicator of the sensor's relevance to the underlying degradation status. Therefore, we only remove sensors with variance less than 10^{-4}. As a result, 14 sensors are preselected from a total of 21 sensors. The preselected sensors are: T24, T30, T50, P30, Nf, Nc, Ps30, Phi, Nrf, Nrc, BPR, htBleed, W31, W32.

4.4.2 RUL Prediction Results

First, we report the sensor selection results. Our proposed sensor selection method indicates that 12 out of the 14 sensors (excluding sensors Nc and Nrc) are informative. The first observation from the sensor selection results is that all sensors associated with the HPC (i.e., T30, P30, Ps30, phi) have all been labeled as informative. This aligns with our prior understanding as the FD001 dataset used in the case study only contains failures associated with the HPC. The second observation is that the remaining 8 informative sensors (i.e., T24, T50, Nf, Nrf, BPR, htBleed, W31, and W32) have also been labeled as important sensors in multiple past research. Although the relationship of the 8 informative sensors to the HPC is not as clear as the previous 4 sensors, these sensors all influence the key control modules (e.g., Low-Pressure Compressor, High-Pressure Turbine, Low-Pressure Turbine) that are closely correlated to the wear and tear of the HPC. Finally, the last observation is that the 2 uninformative sensors (i.e., Nrc and Nc) measure the speed of the core, which is the rotational speed of the central components within the engine. This result also aligns with previous research [4], [74], [75], potentially suggesting that these sensors do not offer critical insights for predicting HPC failures.

Next, we compare the RUL prediction accuracy across the 100 in-service units. Since the in-service units are truncated at random time points, we compare the RUL prediction errors at different levels of actual RUL in Figure 4.9. For instance, "20" on the x-axis represents the in-service units with actual RUL levels equal to or less than 20. The y-axis contains the relative RUL prediction error, which is defined as such:

$$\text{Prediction error} = \frac{|\widehat{RUL} - RUL|}{\text{Actual Failure Time}}.$$

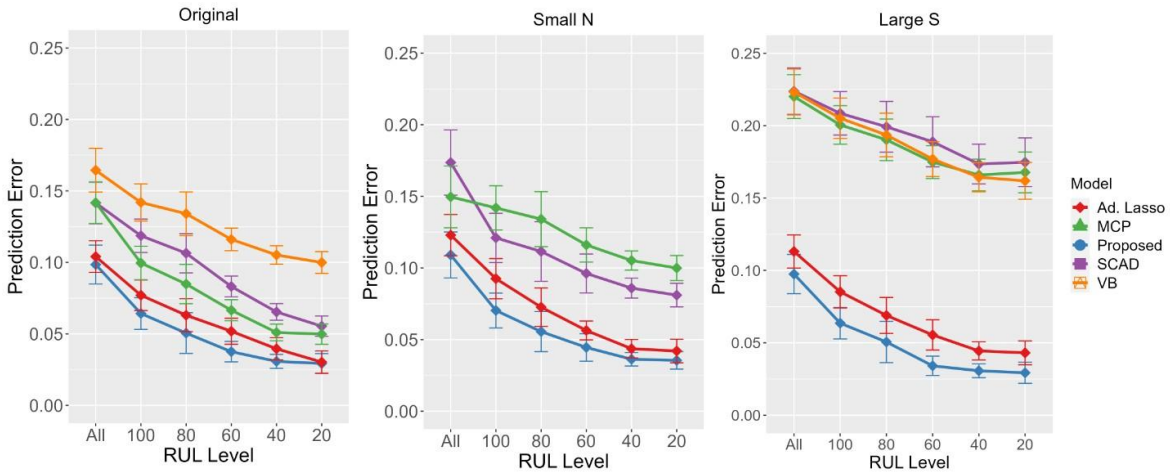


Figure 4.8 (Left) Averaged RUL prediction error results on the C-MAPSS dataset by training on the full 100 training units. (Center) Averaged RUL prediction error on a high-dimensional scenario by training on 15 randomly sampled training units. Number of sensors is untouched. (Right) Averaged RUL prediction error on another high-dimensional scenario by adding 86 randomly generated sensors. Number of training units is untouched. The performance of the proposed method and 4 other benchmark methods are shown. The VB (orange) model for the center plot is omitted due to its significantly poor performance.

Here, $\widehat{RUL}$ refers to the predicted RUL, while RUL represents the true RUL of the in-service unit. Note that in Figure 4.8, the solid points represent the average prediction errors across each RUL level, and the error bars represent a single standard deviation of the errors. Results in the leftmost plot show that the proposed spike-and-slab approach yields the lowest prediction error across all levels of actual RUL. Please note that even though the original C-MAPSS dataset is not very high dimensional, the proposed method still manages to outperform benchmark approaches. The constructed HI using the proposed method and the 12 informative sensor signals are shown in Figure 4.9.

4.4.3 Results Under High-dimensional Scenarios (small N)

In this subsection, we mimic a high-dimensional scenario by randomly reducing the number of

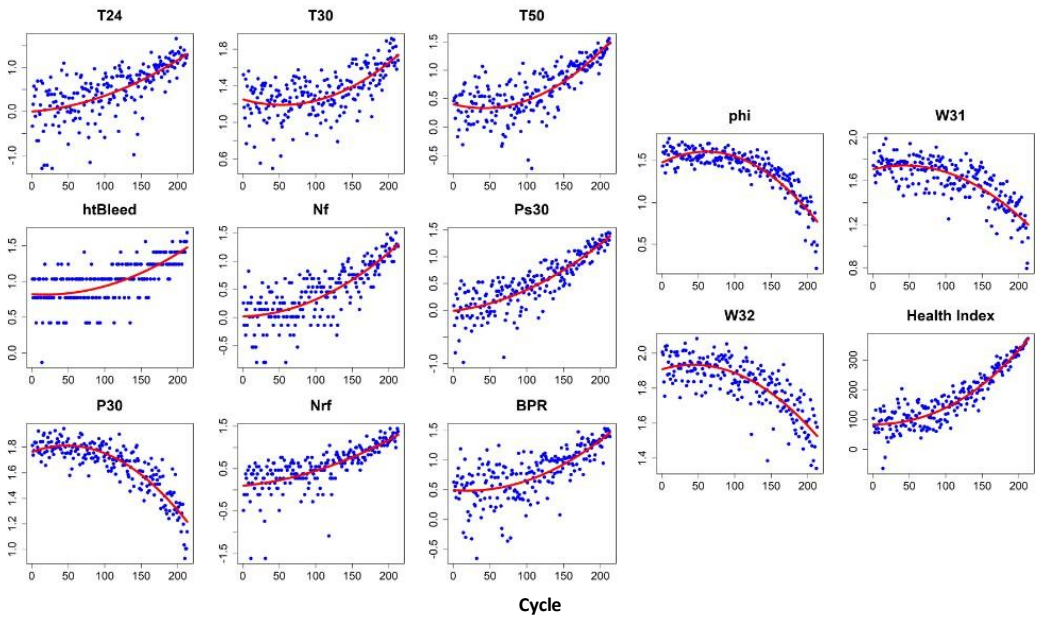


Figure 4.9 Sensor plots of the 12 informative sensors and the constructed HI by the proposed method for a randomly selected in-service unit.

training units. Specifically, a fixed proportion of the 100 training units are sampled. The sampled training units are then used to retrieve the fusion coefficients. Then, similar to Section 4.4.2, we evaluate the RUL prediction accuracy on the 100 in-service units. To properly simulate a high-dimensional scenario, we sampled 15 units out of the 100 training units. Note that the number of training units (i.e., 15) is close to the number of sensors (i.e., 14). The RUL prediction results of this scenario are shown in the center plot of Figure 4.8, which are averaged across 200 iterations to eliminate the effect of random subsampling.

The evaluation results show that the proposed method returns the most accurate RUL predictions. In addition, while other sensor selection methods return higher prediction errors due to the lower number of training units, the proposed method maintains a similar level of performance to the full dataset with 100 training units. Note that the sensor selection results using our proposed method remain the same as in Section 4.4.2.

4.4.4 Results Under High-dimensional Scenarios (large s)

Finally, we impose a different high-dimensional scenario with a very small portion of informative sensors. In particular, we keep the number of training units intact at 100 but introduce additional randomly generated uninformative sensors to further complicate the sensor selection process and RUL prediction. The uninformative sensors are generated using a polynomial mixed effects model with $M = 3$ such that $\boldsymbol{\psi}(t) = [1, t, t^2] \in \mathbb{R}^{3\times 1}$. The main difference is that the uninformative sensors are assumed to have higher noise levels with more variation. Recall that the set of informative sensors is denoted by $\mathcal{I}$ and the set of uninformative sensors is denoted by $\mathcal{U}$. The cardinality of each set is noted by $|\cdot|$ such that $s = |\mathcal{I}| + |\mathcal{U}|$. The detailed uninformative sensor generation process is listed below:

1. Fit a polynomial regression for each informative sensor $j \in \mathcal{I}$ in the training set and obtain the degradation coefficients $\hat{\boldsymbol{\Gamma}}_j$.
2. Using the residuals of the polynomial regression, obtain the estimated standard deviation $\hat{\sigma}_j^2$ for all informative sensors $j \in \mathcal{I}$. Then, calculate the average noise value via $\mu_{\sigma^2} = \frac{1}{|\mathcal{I}|}\sum_{j=1}^{|\mathcal{I}|} \hat{\sigma}_j^2$
3. If the informative sensor j is decreasing, then we multiply -1 to the coefficients such that $\hat{\boldsymbol{\Gamma}}_j^* = -1 \times \hat{\boldsymbol{\Gamma}}_j$. If the informative sensor j is increasing, we leave it be $\hat{\boldsymbol{\Gamma}}_j^* = \hat{\boldsymbol{\Gamma}}_j$.
4. Calculate the mean and variance of the degradation coefficients $\hat{\boldsymbol{\Gamma}}_j^*$ such that $\boldsymbol{\mu}_\Gamma = \frac{1}{|\mathcal{I}|}\sum_{j=1}^{|\mathcal{I}|} \hat{\boldsymbol{\Gamma}}_j^* \in \mathbb{R}^{3\times 1}, \boldsymbol{\Sigma}_\Gamma = \frac{1}{|\mathcal{I}|-1}\sum_{j=1}^{|\mathcal{I}|}\left(\hat{\boldsymbol{\Gamma}}_j^* - \boldsymbol{\mu}_\Gamma\right)\left(\hat{\boldsymbol{\Gamma}}_j^* - \boldsymbol{\mu}_\Gamma\right)^T \in \mathbb{R}^{3\times 3}$.
5. Sample $\boldsymbol{\Gamma}_{j'} \sim MVN(2\boldsymbol{\mu}_\Gamma, \boldsymbol{\Sigma}_\Gamma)$ for all uninformative sensors $j' \in \mathcal{U}$.
6. Then, we generate the uninformative signals $j' \in \mathcal{U}$ such that $L_{i,j'}(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_{j'} + \varepsilon_{i,j'}(t)$ for all training units $i = 1, \dots, N$, where $\varepsilon_{i,j'}(t) \sim \mathcal{N}(0, 2\mu_{\sigma^2})$, $t = t_{i,1}, \dots, t_{i,n_i}$, and $\boldsymbol{\psi}(t) =$

$[1, t, t^2] \in \mathbb{R}^{3\times 1}$.

7. Finally, we simulate both increasing/decreasing uninformative sensors by sampling $\xi_{j'}$~Unif(0,1), where $\begin{cases} L_{i,j'}(t) = L_{i,j'}(t) \text{ if } \xi_{j'} \geq 0.5 \\ L_{i,j'}(t) = -L_{i,j'}(t) \text{ if } \xi_{j'} < 0.5 \end{cases}$ for all $j' \in \mathcal{U}$.

8. Repeat step 6 for the testing units. Ensure that uninformative sensors of the testing units are generated with the same degradation coefficients $\boldsymbol{\Gamma}_{j'}$ and trend values $\xi_{j'}$.

The generated uninformative sensors have more variation due the average noise term μ_{σ^2} and the mean of the degradation coefficients $\boldsymbol{\mu}_{\Gamma}$ are multiplied by 2 (i.e., $2\mu_{\sigma^2}$ in step 6 and $2\boldsymbol{\mu}_{\Gamma}$ in step 5). Note that both informative and uninformative signals still display monotonic behavior, so it is not possible to apply heuristic methods to screen out informative sensors. A sample plot of the simulated uninformative sensor with comparison to sensor T24 is shown in Figure 4.10. For the evaluations, we generate 86 uninformative sensors, resulting in 100 total sensors. We repeat the iterations 200 times and record the RUL prediction results in the rightmost plot of Figure 4.8.

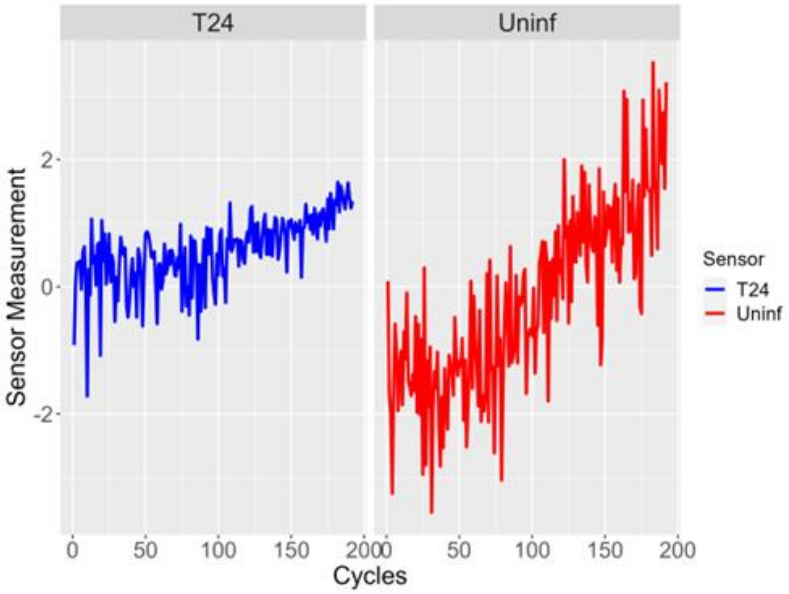


Figure 4.10 Degradation signal plots for the T24 sensor (blue) versus a simulated uninformative sensor (red) for a random training unit

The results again show that the proposed model maintains the best RUL prediction performance even under the contamination of uninformative sensors. Furthermore, the sensor selection results remain unchanged as in Section 4.4.2, demonstrating the robustness of the proposed sensor selection approach. During the evaluations, we also discovered that the proposed model properly concluded that the 86 simulated sensors are uninformative for 198 out of the 200 iterations. Hence, the proposed model still retains effective prognostic results.

4.5 Conclusion

In this paper, we proposed a novel data-fusion method tailored to high-dimensional sensor scenarios for better prognostics. Specifically, the proposed method uses a spike-and-slab prior distribution on the fusion coefficients that automatically selects the informative sensors, which are then fused into a 1-D HI for RUL prediction. The proposed method has the following unique advantages. First, the proposed spike-and-slab sensor selection approach significantly outperforms existing sensor selection methods, especially under high-dimensional scenarios with many sensors relative to the number of training units. Second, the proposed method also boasts superior sensor selection performance even under the influence of sensor correlation. Third, the proposed methods have nice theoretical properties like weak and strong selection consistency. Finally, the proposed method demonstrates high RUL prediction accuracy relative to existing benchmark methods.

The prognostic performance under different scenarios was meticulously investigated through the simulation and case studies. The simulations in Section 4.3 investigated the sensor selection performance with and without correlation, with results showing reliable sensor selection performance even under high cross-correlation. For the case study, evaluations on the C-MAPSS dataset demonstrated the superior RUL prediction performance of the proposed method. Even under high-dimensional scenarios such as lower number of training units and contamination of

uninformative sensors, the proposed method still showed the lowest prediction errors relative to existing benchmark methods.

There are several interesting topics for future research. First, we assumed a single failure mode and operating condition. In future studies, we aim to propose a more generic sensor selection approach for multiple failure modes and operating conditions under the high-dimensional settings. For instance, we can extend the current method by introducing a latent variable for each failure mode and then treat this latent variable as an additional parameter with its own distribution and integrate it to the Bayesian parameter estimation. Second, we implicitly assume that the set of informative sensors does not change with respect to time. However, it is possible for the set of informative sensors to change/evolve as the system degrades, especially when there are multiple failure modes and operation conditions. In the future, we aim to extend this work into an adaptive sensor selection framework so that the model can select a different set of sensors with respect to the current degradation status.

4.6 Appendix

In this appendix, we compare the proposed method with a deep learning-based sensor selection approach by Kim et al. [89]. The sensor selection results were recorded across three scenarios of the case study (i.e., Original, small N, and large s) in Table 4.4 of Section 4.4. Results show that the deep learning approach [89] is highly sensitive to the different high-dimensional conditions

Table 4.4 Sensor selection performance of a deep learning-based approach by Kim et al. [88] on the C-MAPSS FD001 dataset

Scenarios	Selected Informative Sensors
Original	T24, T50, P30, Nf, Ps30, phi, Nrf, htBleed, W32 (9 sensors)
Small N	T24, T50, P30, Ps30, htBleed, W32 (6 sensors)
Large s	T50, P30, Ps30, htBleed, W32 (5 sensors)

and selects an inconsistent number of informative sensors (9,6,5) for each scenario. On the contrary, our proposed method has maintained stable sensor selection results (i.e., 12 informative sensors) across all three scenarios.

Chapter 5 An uncertainty-informed neural network-based (UINN) prognostic model for multi-type data

5.1 Introduction

Recent advances in sensor technology have sparked the widespread use of multiple sensors to monitor the system's condition. These sensors collect key operational parameters, which are then used for various analytics tasks such as anomaly detection, remaining useful life (RUL) prediction, and scheduling appropriate maintenance actions. The data collected from these sensors can be broadly divided into two categories: continuous sensor signals and discrete event sequences. Each data type provides unique insights into the unit's health status. Discrete event data typically captures sudden changes such as anomalies or maintenance actions, whereas continuous signal data highlights the gradual changes and temporal trends within the system. For instance, a typical manufacturing equipment has a group of sensors recording the various events (e.g., maintenance, operational condition changes) in the system, and another group of sensors recording changes in mechanical parameters of the system (e.g., vibration, temperature, pressure).

For accurate prognostics, it would be ideal to draw prognostic insights from both data types and integrate them into the final RUL predictions. However, existing works tend to focus on analyzing a single data type. There has been a plethora of research on predicting the RUL by only analyzing continuous sensor signals [10], [39], or discrete event data [42], [103], but very few works have tried to simultaneously incorporate both data types into the final RUL predictions. A few works in reliability and statistics literature have investigated joint modeling of both data types [104], but their assumptions on the events are very restrictive as the events are generally failure/terminal

univariate events. Such restrictive assumptions greatly limit the applicability of these joint modeling approaches, as the unit can repeatedly experience a wide range of non-terminal events that affect its health condition.

One possible approach for processing both data types is to manually extract features from both data types and then use them for RUL prediction. However, this process is labor-intensive and requires a deep understanding of the intricate relationship between events, sensor signals, underlying degradation status, and RUL. Due to this complex structure, it can be extremely difficult to formulate a generalizable feature extraction procedure across many applications. Another possible approach is to formulate a parametric statistical model for each data type and its relationship with the RUL. But this approach is neither practical nor scalable, as the model structure can become exponentially complex with assumptions on the event type (i.e., recurrent, multi-type) and the relationship between the degradation status and the event/signal data.

Alternatively, deep learning (DL) approaches have recently gained great popularity due to their strong performance in a wide range of applications in healthcare [48] and prognostics [105]. In addition to their outstanding predictive performance, another major advantage of DL approaches for prognostics is their ability to directly learn the intricate dynamics of complex engineering systems from the available data. As a result, DL approaches do not require extensive feature engineering efforts and can automatically extract relevant features from both continuous sensor signals and discrete event data. Despite the success of DL approaches, direct applications of existing DL models to prognostics may not yield satisfactory performance. First, the flexible architecture of DL models can sometimes lead to challenging modeling issues. When designing a DL model, one needs to choose multiple hyperparameters such as the number of hidden layers, optimizers, learning rates, and activation functions. Moreover, developing a DL model for

different data types necessitates specific model structures. For instance, while both event data and signal data exhibit temporal trends, categorical data like event types typically require additional embedding layers to convert them into vector representations. The design complexity increases further when integrating the insights from each data type into the final RUL predictions. Therefore, formulating a DL prognostic model for multi-type data requires careful model design choices. Second, even with a well-designed model architecture, training the model comes with its own set of challenges. To prevent unwanted bias, the standard approach involves jointly training separate predictors for each data type. However, joint training is difficult as the DL model can easily experience over/underfitting issues by failing to efficiently learn features from all data types at equal rates [6]. Without proper procedures, it is common for DL models to focus on learning one dominant data type, resulting in imbalanced learning and suboptimal performance.

Based on these challenges, the objective of this paper is to formulate a data-driven, DL-based prognostic model that leverages insights from both discrete event data and continuous signal data. The proposed model is referred to as the uncertainty-informed neural network (UINN) model. The key contributions of this work are summarized as follows. First, to the best of our knowledge, the proposed UINN model is the first DL prognostic model that simultaneously captures the dynamics of discrete event data and continuous signal data. As a result, the proposed model can provide a holistic picture of the underlying degradation status compared to analyzing a single data type. Second, using a DL model avoids the need for restrictive parametric assumptions. Unlike statistical models, the proposed model can accommodate a wide range of event interactions, multi-type, and recurrent events. Third, to overcome the training challenges, the UINN model presents a joint training procedure to minimize estimation bias and achieve better prognostic performance. Specifically, the UINN model uses a joint loss function that is a weighted sum of the loss

components corresponding to each prediction task: event type, event time, signal, and RUL prediction. The weights of each loss component are determined by the uncertainty information of each prediction task, with higher weights given to tasks with lower uncertainty. Evaluation results on simulated data and real-life case study data show that the UINN model outperforms existing benchmarks. In particular, the case study includes a new battery discharge data using the PiSugar battery attached to a Raspberry Pi device. Details of the data collection and the hardware used are provided in Section 5.4.2.1. This new battery dataset will also serve as a valuable resource for various other prognostic studies related to multi-type data analysis and battery research.

The rest of this paper is organized as follows. Section 5.2 provides a review of existing prognostic techniques for modeling discrete event data, analyzing multi-type data, as well as references for leveraging uncertainty information to alleviate training issues. Section 5.3 describes the details of the proposed UINN prognostic model, including the structure of each predictor and the joint training procedure. Then, Section 5.4 presents two numerical studies, including a simulation study with generated data and a case study using real-life data collected from PiSugar batteries. Finally, Section 5.5 presents an overall summary and conclusion of the work.

5.2 Literature Review

Literature on event modeling can be broadly categorized into two main branches: statistics-based methods and DL methods. In both branches, events are typically assumed to be either *terminal*, where the system fails after the event occurrence, or *non-terminal* (i.e., *recurrent*), where the system is still functional after the event occurrence. In reliability literature, numerous statistical approaches have been developed to model the occurrence of terminal events. These models primarily rely on techniques from survival analysis, where the time to the terminal event is modeled using popular models such as the Cox proportional hazards (PH) model [42]. Typically,

these models assume a pre-specified parametric relationship between the hazard function (i.e., the instantaneous probability of a terminal event) with a group of covariates (also known as predictors). After the model parameters are estimated, one can plug in the covariates to obtain the mean time-to-failure. Many extensions of the Cox PH models have been proposed, with the most prominent one being joint models. First proposed by [56], joint models draw prognostic insights from both continuous longitudinal data (i.e., degradation signals) and time-to-event data. Normally, the degradation signals are modeled by a mixed-effects model. Then, the fitted signals are plugged into the Cox PH model as covariates to compute the corresponding hazard and survival probabilities. These joint models have shown promising results in both medical and prognostic applications. Recent advancements have introduced joint models that use multivariate gaussian processes [44] for greater modeling flexibility and prediction accuracy.

While these models offer strong predictive capabilities, their reliance on parametric relationships significantly limits their flexibility. A key limitation of the Cox model is the assumed linear relationship between the covariates and the log-hazard function. In practice, these quantities can have complex, nonlinear relationships that the Cox PH model cannot effectively capture. Another limitation arises from the direct use of longitudinal observations (e.g., degradation signals) as time-varying covariates, which introduces two major sources of bias in the estimation process. First, the parameter estimates are biased due to "Last-Observation-Carried-Forward" (LOCF) inference method [106], where the most recent observation is used instead of the true time-dependent covariates. Second, measurement errors in the longitudinal observations introduce additional bias into the inference process [107]. Finally, these approaches are designed to only handle terminal events. In practice, the system can experience a wide range of non-terminal, multi-

type, recurrent events that affect the underlying degradation status. As a result, these methods have limited applicability in real-world scenarios.

Traditional statistics-based approaches for modeling recurrent events employed renewal processes [108]. These approaches typically assume that the system is fully restored to a 'healthy' state following maintenance actions. However, in practice, maintenance can have varying effects on the underlying system and does not always fully restore system health. To address this limitation, [109] proposed a class of imperfect maintenance models that apply a geometric reduction to the system's age or event intensity. [110] proposed a multi-type recurrent event model for multi-component systems with imperfect maintenance actions. In general, although there are many statistical approaches that accommodate multi-type recurrent events, they still impose strong parametric assumptions on the degradation trends of sub-systems by requiring the user to select the appropriate baseline process. In addition, these statistical approaches often struggle to scale with the number of event types.

Another group of statistics-based approaches to model recurrent events are called temporal point processes (TPP), which are probabilistic generative models that capture the dynamics in event sequences [111]. TPPs do share some similar model structures with survival models for recurrent events, but the main difference is that survival models are interested in predicting the time to a terminal event, while TPPs focus on modeling the intensity of recurrent event occurrences over time. A general review of TPPs and their theoretical foundations can be found in [112]. However, a major limitation of classical TPPs is their parametric nature. Like survival models, they require the user to specify a baseline intensity and an intensity function, which can restrict the model's ability to capture complex event dependencies.

In general, the main limitation of statistics-based approaches is their focus on parametricity, which restricts their ability to capture a wide range of complex interactions. To address this limitation, researchers have increasingly turned to DL approaches. By replacing the parametric functions with neural networks, these approaches offer much greater flexibility in modeling intricate functional relationships. For modeling the occurrence of terminal events, researchers have proposed to extend the Cox model using neural networks by replacing the linear covariate term with a generic neural network. Due to their flexibility, these models have outperformed traditional survival models in a variety of clinical and reliability applications, e.g., DeepSurv [48] and SurvivalNet [113].

Neural network approaches have also been used to model the occurrence of non-terminal, recurrent actions, especially in the context of TPPs. Instead of specifying a parametric form of the event intensity, researchers have used a neural network to parameterize the intensity function. These NN-based extensions are commonly referred to as neural TPPs and have gained popularity in recent years due to their great predictive power and flexibility. The seminal work of [114] provided a general framework of neural TPPs, where each event is first represented as a feature vector. The sequence of feature vectors is then encoded into a fixed-dimensional history embedding vector, which is then used to derive the conditional distribution over the next event. Many variants have been proposed, including which information to include in the feature vector [115] and how to effectively encode the event history into a fixed-dimensional vector [116]. However, a common limitation of these approaches is the difficult training process due to their intricate model structures. In fact, empirical results have shown that neural TPPs are more susceptible to fitting issues and are highly sensitive to the choice of various model components [111]. Therefore, training a neural TPP often necessitates extensive efforts, including multiple

cross-validation steps, regularization strategies, and large-scale data collection to ensure adequate generalization performance.

In summary, there are two main literature gaps that need to be addressed. First, there is a lack of data-driven methods that integrate insights from both discrete event data and continuous longitudinal data. Although there are some models (i.e., joint models) that capture the effect of both data types, they tend to be focused on terminal events and not on recurrent, non-terminal events. In practice, non-terminal events, such as maintenance actions, have a significant effect on the underlying degradation status. Therefore, the effect of non-terminal events must be accounted for in the model. Second, DL approaches are still difficult to train, with many of them frequently encountering model fitting issues. To avoid these common pitfalls and fully exploit the predictive power of neural networks, there is a need for an established, systematic training procedure for these models. In response, the proposed UINN model is a flexible, DL approach that: 1) accounts for the effect of multiple recurrent, non-terminal events as well as longitudinal signals; and 2) uses a systematic, joint training procedure based on uncertainty information to assist model training and accelerate the convergence of the loss function. The details of the UINN model are introduced in Section 5.3.

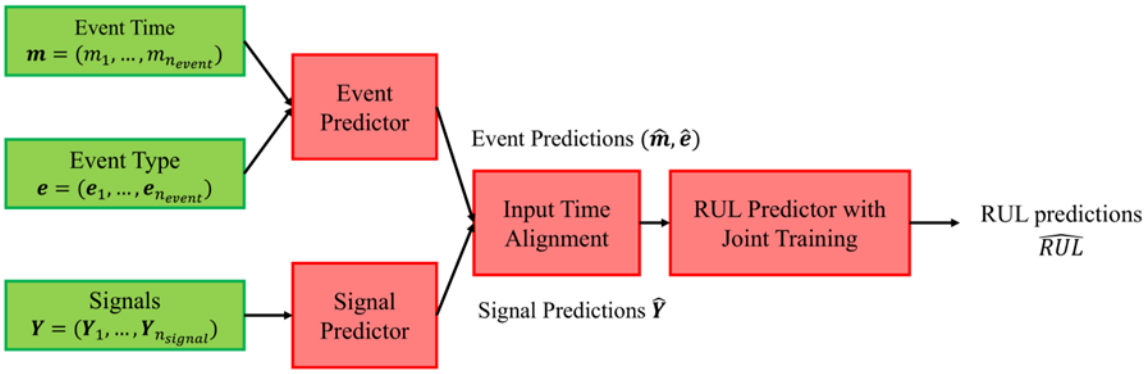


Figure 5.1 Architecture of the proposed UINN model (Notation on unit i is dropped for convenience).

5.3 Methodology

This section describes the details of the proposed UINN model for multi-type data. An overview of the UINN model is shown in Figure 5.1 with three major components: the event predictor, the signal predictor, and the RUL predictor. As the name suggests, the event predictor processes the discrete event data and predicts the next event type and time, while the signal predictor takes the continuous multivariate signal data and predicts the subsequent set of signals. Then, the predicted event and signal data are fed into the RUL predictor for the final RUL prediction. Since each predictor has varying data types as inputs, each predictor has its own unique structure. Section 5.3.1 first discusses the problem formulation. Section 5.3.2.1 describes the details of the event predictor, Section 5.3.2.2 illustrates the details of the signal predictor, and Section 5.3.2.3 describes the data alignment process used to prepare the predicted event and signal data as inputs for the RUL predictor. Finally, Section 5.3.3 describes the joint training procedure to fit the UINN model and explains how uncertainty information aids the training process by automatically weighing the loss components of each predictor.

5.3.1 Problem Formulation

Suppose that there are N_{train} historical units that produce event sequence data and degradation signal data. In particular, a given unit $i \in \{1, \dots, N_{train}\}$ has associated event sequence data $\boldsymbol{E}_i$ and degradation signal data $\boldsymbol{Y}_i$. The event data $\boldsymbol{E}_i = \{\boldsymbol{e}_{i,j}, m_{i,j}\}_{j=1}^{n_{i,event}}$ has two main components: the event type $\boldsymbol{e}_{i,j}$, expressed as a one-hot encoded vector $\boldsymbol{e}_{i,j} = (e_{i,j,1}, \dots, e_{i,j,Z}) \in \mathbb{R}^{1\times Z}$, where the zth position (for $z \in \{1, \dots, Z\}$) is 1 for the zth event type and all other positions are 0, and the associated event time (i.e., time of event occurrence) $m_{i,j}$ for the jth event in the sequence. Note that $n_{i,event}$ represents the total number of events in the event sequence for unit i.

The events considered in this study have three major characteristics: *multivariate*, *recurrent*, and *non-terminal*. First, multivariate means that there are $Z \geq 1$ unique event types in the sequence. Second, recurrent implies that the units can experience the same event type multiple times during their lifetime. Third, the events are non-terminal (referred to as *trigger events* in some literature [42]), so the occurrence of these events can influence the underlying degradation status of the unit but does not indicate that the unit has failed. Examples of non-terminal events are periodic maintenance activities, incorrect machine setup or operation by an operator, early warning diagnostics, and minor faults or errors.

The degradation signals of unit i are represented as $\boldsymbol{Y}_i = \left\{\boldsymbol{Y}_{i,1}, \dots, \boldsymbol{Y}_{i,n_{i,signal}}\right\}$. Each $\boldsymbol{Y}_{i,j} \in \mathbb{R}^p$ term contains signal observations from p sensors at the jth observation, where $j = \{1, \dots, n_{i,signal}\}$. The respective observation times for the degradation signals are denoted as $t_{i,1}, \dots, t_{i,n_{i,signal}}$. Next, the corresponding RUL values of unit i are denoted as $\boldsymbol{RUL}_i = \left\{RUL_{i,1}, \dots, RUL_{i,n_{i,signal}}\right\}$. Notice that due to the characteristics of physical sensors, the degradation signals and the RUL values are measured at the same time grids $t_{i,1}, \dots, t_{i,n_{i,signal}}$, which is different from the time grid of the events $\boldsymbol{m}_i = \{m_{i,1}, \dots, m_{i,n_{event}}\}$. This is expected because events usually occur at irregular intervals, so the number of signal or RUL observations $n_{i,signal}$ is not equal to the number of event observations $n_{i,event}$ (i.e., $n_{i,signal} \neq n_{i,event}$). Since neural networks require inputs to be aligned on the same time grids, Section 5.3.2.3 introduces an input alignment procedure for the event, signal, and RUL values in the proposed UINN model. Once the events and signals are aligned on the same time grid, they are fed into the UINN model to predict the future event times, types, degradation signals, and RUL.

5.3.2 Proposed Network for Multi-type Data

This subsection describes the structure of the proposed UINN model. As illustrated in Figure 5.1, each data type requires a separate predictor to capture its unique characteristics. Details of each predictor (i.e., event predictor, signal predictor, and the RUL predictor with the input alignment step) are discussed below.

5.3.2.1 Event Predictor

The temporal dynamics of the event sequence $\boldsymbol{E}_i = \{(\boldsymbol{e}_{i,j}, m_{i,j})\}_{j=1}^{n_{i,event}}$ is captured via a variant of the Long Short-Term Memory (LSTM) model. Before plugging the event sequence into the event predictor, it is preprocessed using a sliding window approach, where a fixed window width (i.e., number of events) TW_{event} is applied. Each input instance after the sliding window approach is denoted as $\boldsymbol{E}_{i,s} = \{(\boldsymbol{e}_{i,j}, m_{i,j})\}_{j=s-TW_{event}+1}^{s}$ for $s \in \{TW_{event}, TW_{event}+1, \dots, n_{i,event}-1\}$. Then, the objective of the event predictor is to predict the next event type $\boldsymbol{e}_{i,s+1}$ and event time $m_{i,s+1}$.

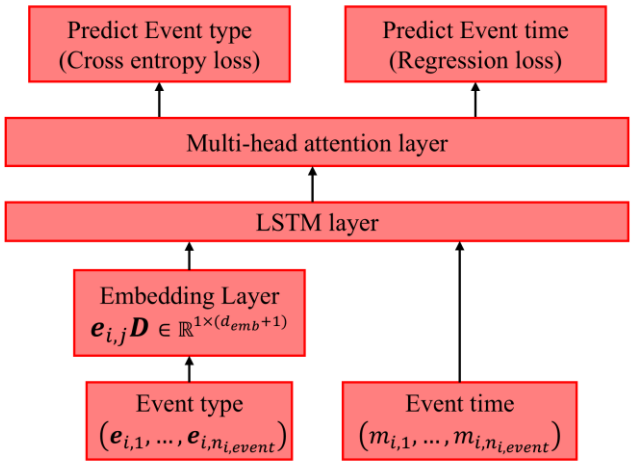


Figure 5.2 Detailed architecture of the event predictor.

The overall structure of the event predictor is illustrated in Figure 5.2 with the detailed input-output procedure explained in the following:

1. The Z-dimensional one-hot event type vector $\boldsymbol{e}_{i,j} \in \mathbb{R}^{1\times Z}$ is first passed through an embedding matrix $\boldsymbol{D} \in \mathbb{R}^{Z\times d_{emb}}$, where d_{emb} is the embedding dimension.
2. The transformed event type vector $\boldsymbol{e}_{i,j}\boldsymbol{D} \in \mathbb{R}^{1\times d_{emb}}$ is concatenated with the event time $m_{i,j}$ such that $\left[\boldsymbol{e}_{i,j}\boldsymbol{D}, m_{i,j}\right] \in \mathbb{R}^{1\times(d_{emb}+1)}$.
3. This concatenated vector from Step 2 is fed into the LSTM layer, LSTM_{event}, which produces a sequence of hidden states $\boldsymbol{H}_{i,s}^{event} = \text{LSTM}_{event}\left(\left\{\boldsymbol{e}_{i,j}\boldsymbol{D}, m_{i,j}\right\}_{j=s-TW_{event}+1}^{s}, \boldsymbol{H}_{i,s-1}^{event}, \boldsymbol{c}_{i,s-1}^{event}\right) \in \mathbb{R}^{TW_{event}\times d_{hidden}}$. Note that d_{hidden} is the dimension of the LSTM hidden states, $\boldsymbol{H}_{i,s-1}^{event} \in \mathbb{R}^{TW_{event}\times d_{hidden}}$ is the hidden state of the event predictor at time $m_{i,s-1}$, and $\boldsymbol{c}_{i,s-1}^{event} \in \mathbb{R}^{TW_{event}\times d_{hidden}}$ is the cell state of the event predictor at time $m_{i,s-1}$. The difference between the two states is that the cell state pays more attention to long-term dependencies, while the hidden state focuses on short-term dependencies.
4. The hidden states $\boldsymbol{H}_{i,s}^{event}$ are passed through a multi-head attention layer to improve the model's ability to capture long-range dependencies in the event sequence. The resulting vector is represented as $\boldsymbol{U}_{i,s}^{event} = \text{MultiHeadAttention}\left(\boldsymbol{H}_{i,s}^{event}\right) \in \mathbb{R}^{TW_{event}\times d_{hidden}}$.
5. The latest observation of $\boldsymbol{U}_{i,s}^{event}$, defined as $\boldsymbol{u}_{i,s}^{event} = \boldsymbol{U}_{i,s}^{event}[TW_{event}, :] \in \mathbb{R}^{1\times d_{hidden}}$, is processed through two separate dense layers to predict the next event type and event time, resulting in $\hat{e}_{i,s+1} = \tanh\left(\boldsymbol{u}_{i,s}^{event}\boldsymbol{W}_{type} + b_{type}\right)$ and $\hat{m}_{i,s+1} = \tanh\left(\boldsymbol{u}_{i,s}^{event}\boldsymbol{W}_{time} + \boldsymbol{b}_{time}\right)$. Here, $\boldsymbol{W}_{type}, \boldsymbol{W}_{time} \in \mathbb{R}^{d_{hidden}\times 1}$ represent the weights of the event type and event time layers, and $b_{type}, b_{time} \in \mathbb{R}^{1\times 1}$ are the corresponding bias

terms. Note that the event predictor returns a scalar prediction $\hat{e}_{i,s+1}$ instead of the one-hot vector $\hat{\boldsymbol{e}}_{i,s+1}$.

The multi-head attention layer [117] used in step 4 is a mechanism that allows the model to better capture temporal dependencies across different segments in the input sequence. Past research has shown that adding attention layers to LSTMs improves their capability of capturing long-range dependencies in both event sequence and time series modeling [117]. Multi-head attention extends this concept by performing multiple attention functions in parallel and then averaging the results across the attention functions (i.e., heads), allowing the model to even better capture the temporal trends of the sequence.

The next step is to define an appropriate loss function to train the event predictor. Since event type prediction is a classification task and event time prediction is a regression task, the network is trained on a weighted sum of the event time prediction loss and the event type prediction loss. The total loss function for the event predictor is shown below:

$$Event\ Loss = w^{time}\mathcal{L}_{event}^{time} + w^{type}\mathcal{L}_{event}^{type},$$

$$\mathcal{L}_{event}^{time} = \sum_{i=1}^{N_{train}} \sum_{j=TW_{event}+1}^{n_{i,event}} \left(m_{i,j} - \hat{m}_{i,j}\right)^2,\ \mathcal{L}_{event}^{type} = \sum_{i=1}^{N_{train}} \sum_{j=TW_{event}+1}^{n_{i,event}} \mathrm{CE}\left(e_{i,j}, \hat{e}_{i,j}\right),$$

$$\mathrm{CE}\left(e_{i,j}, \hat{e}_{i,j}\right) = \sum_{z=1}^{Z} e_{i,j,z} \log\left(\hat{e}_{i,j,z}\right). \tag{5.1}$$

where $e_{i,j,z}$ is a binary indicator that is 1 if event z is the true event for unit i's jth event, $\hat{e}_{i,j,z}$ is the predicted probability of the jth event being event z, and CE is the cross-entropy function. Details on how to configure the weights of each loss function w^{type} and w^{time} are explained in Section 5.3.3.

5.3.2.2 Signal Predictor

The signal predictor is also based on the LSTM architecture due to the time-dependent characteristics of degradation signals. Given a window size TW_{sig}, the degradation signals $\{\boldsymbol{Y}_{i,j}\}_{j=t-TW_{sig}+1}^{t}$ for $t \in \{TW_{sig}, \dots, n_{i,signal}\}$ are passed into an LSTM model, which returns a vector of hidden states with $\boldsymbol{H}_{i,t}^{sig} \in \mathbb{R}^{TW_{sig} \times d_{hidden2}}$. Here, $d_{hidden2}$ represents the hidden layer size of the LSTM model. Then, the final hidden state $\boldsymbol{h}_{i,t}^{sig} = \boldsymbol{H}_{i,t}^{sig}[TW_{sig} - 1, :] \in \mathbb{R}^{1 \times d_{hidden2}}$ is fed into a linear dense layer with a ReLU activation function. The resulting output is the predicted degradation signal at $t+1$, such that $\widehat{\boldsymbol{Y}}_{i,t+1} = \text{ReLU}\left(\boldsymbol{h}_{i,t}^{sig}\boldsymbol{W}_{sig} + \boldsymbol{b}_{sig}\right)$, where $\boldsymbol{W}_{sig} \in \mathbb{R}^{d_{hidden2} \times p}$ is the weight matrix and $\boldsymbol{b}_{sig} \in \mathbb{R}^{1 \times p}$ is the bias term. Note that the signal predictor is trained using the Mean Squared Error (MSE) loss function shown below:

$$\mathcal{L}_{sig} = \sum_{i=1}^{N_{train}} \sum_{j=TW_{sig}+1}^{n_{i,signal}} \left(\boldsymbol{Y}_{i,j} - \widehat{\boldsymbol{Y}}_{i,j}\right)^2. \tag{5.2}$$

5.3.2.3 RUL Predictor with Input Alignment Step

The predicted event time, event time, and signal predictions serve as inputs for the RUL predictor. However, one challenge arises due to the different time intervals between the events and

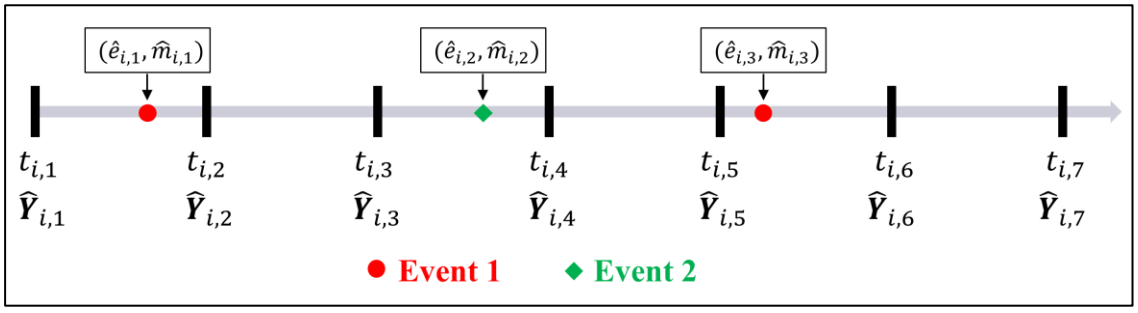


Figure 5.3 Example of misaligned event and time predictions

the degradation signals. As mentioned in Section 5.3.1, degradation signals typically occur at uniform time intervals, but events occur at irregular time intervals. This results in a misalignment between event predictions and signal predictions on the time grid. An example is illustrated in Figure 5.3.

In Figure 5.3, there are signal predictions at each time interval $t_{i,1}, t_{i,2}, \dots, t_{i,7}$. On the contrary, there are three event predictions at irregular time intervals, with the first event occurred between $t_{i,1}$ and $t_{i,2}$, the second occurred between $t_{i,3}$ and $t_{i,4}$, and so on. Since the RUL predictor is a neural network with a fixed input dimension, the predictions must be aligned on the same time grid. Inspired by discrete-time survival analysis models [118], we propose to address this challenge by first dividing the timeframe into disjoint intervals: $(t_{i,1}, t_{i,2}], (t_{i,2}, t_{i,3}], \dots, (t_{i,6}, t_{i,7})$. The time intervals of the signal predictions serve as a reference for the discretization process, as they are often more granular and evenly distributed. A detailed description of the time discretization is shown in Figure 5.4. After discretizing the time grid, Z count variables $\xi_{i,t_1}^1, \xi_{i,t_1}^2, \dots, \xi_{i,t_1}^Z \geq 0$ are defined for each event type. The transformed new inputs are denoted as: $\boldsymbol{X}_{i,2} \in \mathbb{R}^{p+Z} = \text{Concatenate}(\widehat{\boldsymbol{Y}}_{i,2}, \xi_{i,t_2}^1, \dots, \xi_{i,t_2}^Z)$, where $\boldsymbol{X}_{i,2}$ is essentially a concatenated vector of the predicted

New Input $\boldsymbol{X}_t \in \mathbb{R}^{p+Z}$
$\boldsymbol{X}_{i,1}$
$\boldsymbol{X}_{i,2}$
$\boldsymbol{X}_{i,3}$
$\boldsymbol{X}_{i,4}$
$\boldsymbol{X}_{i,5}$
$\boldsymbol{X}_{i,6}$
$\boldsymbol{X}_{i,7}$

=

Time t	**Signals** $\widehat{\boldsymbol{Y}}_t$	**Event 1** $\xi_{i,t}^1$	**Event 2** $\xi_{i,t}^2$
$(0, t_{i,1}]$	$\widehat{\boldsymbol{Y}}_{i,1}$	0	0
$(t_{i,1}, t_{i,2}]$	$\widehat{\boldsymbol{Y}}_{i,2}$	1	0
$(t_{i,2}, t_{i,3}]$	$\widehat{\boldsymbol{Y}}_{i,3}$	0	0
$(t_{i,3}, t_{i,4}]$	$\widehat{\boldsymbol{Y}}_{i,4}$	0	1
$(t_{i,4}, t_{i,5}]$	$\widehat{\boldsymbol{Y}}_{i,5}$	0	0
$(t_{i,5}, t_{i,6}]$	$\widehat{\boldsymbol{Y}}_{i,6}$	1	0
$(t_{i,6}, t_{i,7}]$	$\widehat{\boldsymbol{Y}}_{i,7}$	0	0

Figure 5.4 Aligning the event and signal predictions using grid discretization

degradation signals and the event count variables. For instance, in Figure 5.4, the predicted occurrences of event type 1 are in intervals $(t_{i,1}, t_{i,2})$ and $(t_{i,5}, t_{i,6})$, so $\xi^1_{i,t_2} = \xi^1_{i,t_6} = 1$. Similarly, event type 2 happens in interval $(t_{i,3}, t_{i,4})$, so $\xi^2_{i,t_4} = 1$.

Now that the inputs $\boldsymbol{X}_{i,1}, \boldsymbol{X}_{i,2}, \ldots$ are aligned on the same timeframe with consistent input dimensions, and thus they can be fed into the final RUL predictor. This RUL predictor is a simple feedforward neural network consisting of multiple hidden layers with ReLU activation functions. In practice, the number of hidden layers is determined based on the data. Finally, the RUL predictor is trained using the MSE loss function defined as:

$$\mathcal{L}_{RUL} = \sum_{i=1}^{N_{train}} \sum_{j=TW_{sig}+1}^{n_{i,signal}} \left(RUL_{i,j} - \widehat{RUL}_{i,j}\right)^2. \tag{5.3}$$

5.3.3 Joint Training with Uncertainty-informed Loss Function

The proposed UINN model has many predictors, each parameterized by different neural networks with their own loss functions. Specifically, there is the event predictor with event time loss $\mathcal{L}^{time}_{event}$ and event type loss $\mathcal{L}^{type}_{event}$, the signal predictor with its loss function $\mathcal{L}_{sig}$, and the RUL predictor with the RUL loss function $\mathcal{L}_{RUL}$. Due to this complex setup, training the UINN model is tricky and can easily fall into a local minimum with suboptimal performance. To simplify the training procedure, one possibility is to consider a sequential approach, which involves first training the event predictors and signal predictors separately, then using their predictions to train the RUL model in the final step. However, this two-stage approach is known to introduce unwanted bias in the model estimation, potentially leading to erroneous results. A more desirable alternative is joint training all models in a systematic manner to reduce modeling bias.

To jointly train the UINN model, one needs to find the loss weights (i.e., w^{time}, w^{type}, $w^{sig}, w^{RUL} \geq 0$) for the four loss functions (i.e., components):

$$\mathcal{L}_{joint} = w^{time}\mathcal{L}_{event}^{time} + w^{type}\mathcal{L}_{event}^{type} + w^{sig}\mathcal{L}_{sig} + w^{RUL}\mathcal{L}_{RUL}. \quad (5.4)$$

The importance of each prediction task in the UINN model is reflected by the loss component weights, with larger weights signifying greater importance. Past approaches for weight calculation included naïve approaches like uniform weights or manual tuning by trial-and-error. However, manual tuning of the weights is computationally expensive and does not scale well with model complexity. Hence, a more systematic approach that can automatically learn the weights is preferred.

Multi-task learning [119] provides a solution for this challenge. Specifically, multi-task learning aims to improve learning efficiency and prediction accuracy through the simultaneous optimization of multiple tasks (i.e., loss functions) instead of single tasks. To do so, task-dependent (i.e., homoscedastic) uncertainty information is used for weighing the individual loss components. Empirical results indicate that these uncertainty-informed weights can effectively balance multiple tasks and lead to superior performance than naïve counterparts [6]. In the proposed UINN model, each loss component is treated as an individual task and their task-dependent uncertainty is used as weights. Notably, task-dependent uncertainty measures the relative confidence of each task, and it has been frequently used in prior research [120] to weigh losses in a multi-task learning framework.

To define the task uncertainty-informed loss functions, the likelihood functions for the regression tasks (i.e., RUL, event time, and signal prediction) and the classification task (i.e., for event type prediction) are first studied. For regression tasks, a Gaussian likelihood function is used, where the model output serves as its mean and the observation noise term has a variance σ^2. As

an example, for the RUL predictor, $p\big(RUL_{t+1}\,\big|f_{RUL}(\boldsymbol{X}_t)\big)\sim\mathcal{N}(f_{RUL}(\boldsymbol{X}_t),\sigma_{RUL}^2)$, where f_{RUL} is the RUL predictor and σ_{RUL}^2 is the variance of the gaussian observation noise. Note that the subscript i on the units is dropped for notational simplicity. The negative log likelihood of the model can be written as

$$-\log\Big(p\big(RUL_{t+1}\big|f_{RUL}(\boldsymbol{X}_t)\big)\Big)\propto\frac{1}{2\sigma_{RUL}^2}\|RUL_{t+1}-f_{RUL}(\boldsymbol{X}_t)\|^2+\log\sigma_{RUL}. \tag{5.5}$$

Notice that the $\|RUL_{t+1}-f_{RUL}(\boldsymbol{X}_t)\|^2$ term is identical to the MSE loss function of the RUL predictor $\mathcal{L}_{RUL}$ defined in (5.3). The difference is that the loss function is scaled by $\frac{1}{2\sigma_{RUL}^2}$, which can be regarded as the loss component weight. Intuitively, this means that models with higher task-dependent uncertainty receive lower weights, and vice versa. The additional log term, $\log\sigma_{RUL}$, acts as a regularizer that discourages the noise term from increasing too much. The same approach can be used for other regression tasks like event time prediction and signal prediction.

For a classification task like event type prediction, the equation is similar, but the Gaussian likelihood is replaced by a Boltzmann distribution (i.e., scaled version of the model output passed through a Softmax function). For ease notation, we define the input instances of the event predictor as $\boldsymbol{e}_s=\{\boldsymbol{e}_j\boldsymbol{D}\}_{j=s-TW_{event}+1}^{s}$ and $\boldsymbol{m}_s=\{m_j\}_{j=s-TW_{event}+1}^{s}$ for $s\in\{TW_{event},TW_{event}+1,\dots,n_{i,event}-1\}$. Note that the subscript i on the units is again dropped. The model likelihood can be written as:

$$p\big(e_{s+1}=z\big|f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s),\sigma_{type}^2\big)=\text{Softmax}\left(\frac{1}{\sigma_{type}^2}f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s)\right). \tag{5.6}$$

where $f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s)$ represents the outputs (i.e., event type and time predictions) obtained from the event predictor f_{event}. Taking the negative log likelihood of this expression leads (5.6) to the following expression:

$$-\log p\left(e_{s+1}=z\middle|f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s),\sigma_{type}^2\right)=\frac{1}{\sigma_{type}^2}\left(\mathrm{CE}(e_{s+1},f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s)\right)$$

$$+\log\frac{\sum_{z'}\exp\left(\frac{1}{\sigma_{type}^2}f_{event}^{z'}(\boldsymbol{m}_s,\boldsymbol{e}_s)\right)}{\left(\sum_{z'}\exp\left(f_{event}^{z'}(\boldsymbol{m}_s,\boldsymbol{e}_s)\right)\right)^{\frac{1}{\sigma_{type}^2}}}$$

Here, $f_{event}^{z}(\boldsymbol{m}_s,\boldsymbol{e}_s)$ represents the zth element of the output produced by $f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s)$. The following expression can be further reduced with the approximation: $\left(\sum_{z'}\exp\left(f_{event}^{z'}(\boldsymbol{m}_s,\boldsymbol{e}_s)\right)\right)^{\frac{1}{\sigma_{type}^2}}\approx\frac{1}{\sigma_{type}}\sum_{z'}\exp\left(\frac{1}{\sigma_{type}^2}f_{event}^{z'}(\boldsymbol{m}_s,\boldsymbol{e}_s)\right)$, where the equality holds as σ_{type}^2 approaches 1. This simplification has been widely used in deep learning literature [6] and offers practical benefits by reducing computational complexity. It is particularly effective when the predicted logits are sharply peaked. After the simplification, the loss function can be written as:

$$\begin{aligned}&-\log p\left(e_{s+1}=z\middle|f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s),\sigma_{type}^2\right)\\&\qquad\approx\log\sigma_{type}+\frac{1}{\sigma_{type}^2}\left(\mathrm{CE}(e_{s+1},f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s)\right).\end{aligned}\tag{5.7}$$

As shown in (5.7), the simplification results in an objective function that is easier to optimize. With the loss functions for both regression and classification tasks defined, the uncertainty-informed joint loss function can be formulated. The joint likelihood of the entire network $\mathcal{L}_{joint}$ is expressed as follows:

$$\mathcal{L}_{joint}=\prod_{i=1}^{N_{train}}\left[\prod_{s=TW_{event}}^{n_{i,event}-1}p\left(e_{s+1}=z\middle|f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s),\sigma_{type}^2\right)\cdot p(m_{s+1}\middle|f_{event}(\boldsymbol{m}_s,\boldsymbol{e}_s),\sigma_{time}^2)\right.$$

$$\left.\cdot\prod_{t=TW_{sig}}^{n_{i,signal}-1}p\left(\boldsymbol{Y}_{t+1}\middle|f_{sig}(\boldsymbol{Y}_t),\sigma_{sig}^2\right)\cdot p(RUL_{t+1}|f_{RUL}(\boldsymbol{X}_t),\sigma_{RUL}^2)\right].$$

Taking the negative log likelihood of $\mathcal{L}_{joint}$ results in the following expression:

$$-\log \mathcal{L}_{joint} \propto$$
$$\sum_{i=1}^{N_{train}} \left[\sum_{s=TW_{event}}^{n_{i,event}-1} \left(\frac{1}{2\sigma_{time}^2} (m_{s+1} - \hat{m}_{s+1})^2 + \frac{1}{\sigma_{type}^2} \mathrm{CE}(e_{s+1}, \hat{e}_{s+1}) + \log \sigma_{time} + \log \sigma_{type} \right) + \sum_{t=TW_{sig}}^{n_{i,signal}-1} \left(\frac{1}{2\sigma_{sig}^2} \left(\boldsymbol{Y}_{t+1} - \widehat{\boldsymbol{Y}}_{t+1} \right)^2 + \frac{1}{2\sigma_{RUL}^2} \left(RUL_{t+1} - \widehat{RUL}_{t+1} \right)^2 \right) + \log \sigma_{sig} + \log \sigma_{RUL} \right) \right].$$

This is equivalent to (5.4) with $w^{time} = \frac{1}{2\sigma_{time}^2}$, $w^{type} = \frac{1}{\sigma_{type}^2}$, $w^{sig} = \frac{1}{2\sigma_{sig}^2}$, $w^{RUL} = \frac{1}{2\sigma_{RUL}^2}$ without the regularizer terms. Finally, the regularizer terms for each loss component is added to avoid extreme (i.e., too large) variances, resulting in the final joint loss shown below:

$$-\log \mathcal{L}_{joint} = \frac{1}{2\sigma_{time}^2} \mathcal{L}_{time} + \frac{1}{\sigma_{type}^2} \mathcal{L}_{type} + \frac{1}{2\sigma_{sig}^2} \mathcal{L}_{sig} + \frac{1}{2\sigma_{RUL}^2} \mathcal{L}_{RUL} + \log \sigma_{time} + \log \sigma_{type} + \log \sigma_{sig} + \log \sigma_{RUL}. \quad (5.8)$$

5.4 Numerical Study

This section contains two numerical studies to evaluate the performance of the UINN model: one on simulated data in Section 5.4.1 and the other on real-life battery degradation data collected from the PiSugar battery for Raspberry Pi devices in Section 5.4.2. Specifically, Section 5.4.1.1 discusses the details of the data generation process, while Section 5.4.1.2 introduces an overview of the benchmark methods and the evaluation metrics used in this study. Section 5.4.1.3 then presents the evaluation results on the simulated dataset. Next, Section 5.4.2.1 introduces the experimental setup used to collect the battery degradation data from the PiSugar battery. Finally, Section 5.4.2.2 discusses the performance of the UINN model on the real-life dataset. Details of the average computational time and hyperparameter optimization of the simulation study is discussed in the appendix at Section 5.6.

5.4.1 Simulation Study

5.4.1.1 Overview of Data Generation

Consider a unit that produces both discrete event data and continuous signal data. Since the events are non-terminal and only affects the degradation status, only "soft failure" scenarios are considered (i.e., unit i is considered to have failed once the underlying degradation status $\eta_i(t)$ reaches a specific failure threshold l). Specifically, the failure time T_i is defined as such: $T_i = \arg\min_t \eta_i(t) \geq l$. The underlying degradation status $\eta_i(t)$ is affected by two components: the continuous signals, and the cumulative counts of each event type. The exact relationship is specified as such:

$$\eta_i(t) = \boldsymbol{\alpha}\boldsymbol{Y}_i(t) + \boldsymbol{\beta}\boldsymbol{\phi}_i(t). \tag{5.9}$$

Here, $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_p] \in \mathbb{R}^{1\times p}$ is the coefficient vector for the continuous signals, $\boldsymbol{\beta} = [\beta_1, \dots, \beta_Z] \in \mathbb{R}^{1\times Z}$ is the coefficient vector for the discrete events, and $\boldsymbol{\phi}_i(t) = [\phi_{i,1}(t), \dots, \phi_{i,Z}(t)]^T \in \mathbb{R}^{Z\times 1}$ is the cumulative counts for each event type. For instance, if event type 1 has occurred twice by a given time t, then $\phi_{i,1}(t) = 2$. In this simulation study, we generate 2 degradation signals and 4 event types such that $p = 2, Z = 4$. The coefficients for the signals are set as $\boldsymbol{\alpha} = [0.5, 0.3]^T$, reflecting a moderate association with the underlying degradation status. For the events, the coefficients are set as $\boldsymbol{\beta} = [5, 7, 1, 0.1]^T$. This implies that event types 1 and 2 have strong associations with the underlying degradation status, event 3 has a moderate association, whereas event 4 has a negligible impact.

The signals are generated using a mixed effects model with a polynomial basis function such that $Y_{i,1}(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_{i,1} + \varepsilon_{i,1}(t)$, with $\boldsymbol{\psi}(t) = [1, t, t^2] \in \mathbb{R}^{1\times 3}$ as the basis, and $\boldsymbol{\Gamma}_{i,1} \in \mathbb{R}^{3\times 1}$ as the corresponding degradation coefficient. The degradation coefficients are sampled from a

multivariate normal distribution $\boldsymbol{\Gamma}_{i,1}\sim MVN(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \boldsymbol{\Gamma}_{i,2}\sim MVN(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ with $\boldsymbol{\mu}_1 = [2.5,0.1,0.01]^T, \boldsymbol{\mu}_2 = [1.5,0.1,0.01]^T, \boldsymbol{\Sigma}_1 = \begin{pmatrix} 0.2 & -4e-4 & 7e-5 \\ -4e-4 & 3e-6 & 1e-7 \\ 7e-5 & 1e-7 & 3e-6 \end{pmatrix}, \boldsymbol{\Sigma}_2 = \begin{pmatrix} 0.1 & -2e-4 & 4e-6 \\ -2e-4 & 3e-6 & 1e-8 \\ 4e-6 & 1e-8 & 3e-6 \end{pmatrix}$. Finally, $\varepsilon_{i,1}(t), \varepsilon_{i,2}(t)\sim N(0,1^2)$ are the added Gaussian noise terms.

Simulating the events is more challenging as there are multiple event types with possible correlations among the events. One popular method for event sequence generation is the multivariate Hawkes Process, which is a self-exciting, $Z-$dimensional point process defined by its conditional intensity function $\lambda_z(t)$ shown below:

$$\lambda_z(t) = \mu_z + \sum_{z'=1}^{Z} \sum_{k:t_k^{z'}<t} \varphi_{zz'}\left(t - t_k^{z'}\right).$$

The intensity function $\lambda_z(t)$ indicates the infinitesimal probability of an event z occurring during the time interval $[t, t+dt]$. Notice that the intensity is affected by both the baseline intensity μ_z and the past events (i.e., including other event types) that occurred before time t. The contribution of the past events z' to the intensity of event z is measured by the nonnegative triggering kernel $\varphi_{zz'}$ $(\varphi_{zz'}(t) \geq 0, \forall t \geq 0)$ and the degree of time decay $t - t_k^{z'}$, where $t_k^{z'}$ represent the timestamps of all events of event type z'. For this simulation study, the exponential parameterization of the kernel is used such that $\varphi_{zz'}(t) = \nu^{zz'}\chi^{zz'}\exp\left(-\chi^{zz'}t\right)\mathbb{I}_{t>0}$, where $\left\{\nu^{zz'}\right\}_{z,z'\in Z}$ represents the adjacency matrix (i.e., measures the effect from event z and z') and $\left\{\chi^{zz'}\right\}_{z,z'\in Z}$ represents the decay matrix (i.e., controls the degree of time decay), and $\mathbb{I}_{t>0}$ is the identity function. Note that we fix the decay parameter $\chi^{zz'} = 1$ to simplify the simulation and

focus on the effect of the adjacency matrix and baseline intensity. For all four event types $z \in \{1,..,4\}$, the baseline intensity μ_z is sampled from a uniform distribution such that $\mu_z \sim \text{Unif}(0.05,0.10)$. To simulate a wide range of event interactions, it is assumed that the first event type is influential to all other events, the second and third event types are moderately influential, and the fourth event type has negligible influence. As a result, the adjacency matrix $\{\nu^{zz'}\}_{z,z' \in Z=\{1,2,3,4\}}$ is set such that the diagonal terms $\{\nu^{zz'}\}_{z=z'}$ is set to a baseline level of 0.01. The non-diagonal terms are set to $\{\nu^{zz'}\}_{z \neq z', z=1} \sim \text{Unif}(0.05,0.06)$, $\{\nu^{zz'}\}_{z \neq z', z=2,3} \sim \text{Unif}(0.01,0.02), \{\nu^{zz'}\}_{z \neq z', z=4} = 0$.

Once all continuous signal data and discrete events are simulated, they are plugged into (5.9) to obtain the final underlying degradation status $\eta_i(t)$. Then, the failure times of each unit i is recorded when $\eta_i(t)$ reaches a pre-specified failure threshold l, which is set to 100 in this case. Figure 5.5 shows an example of a sample training unit's degradation status, with the event's occurrence time marked in each color. The difference in the jumps of the underlying degradation status represents the varying levels of influence of each event on the unit's degradation.

Note that this simulation process can be easily extended to accommodate varying number of event types, degradation signals, and failure thresholds. All computations were done in Python 3.9.10, with the Hawkes process simulated using the Python "tick" library [121].

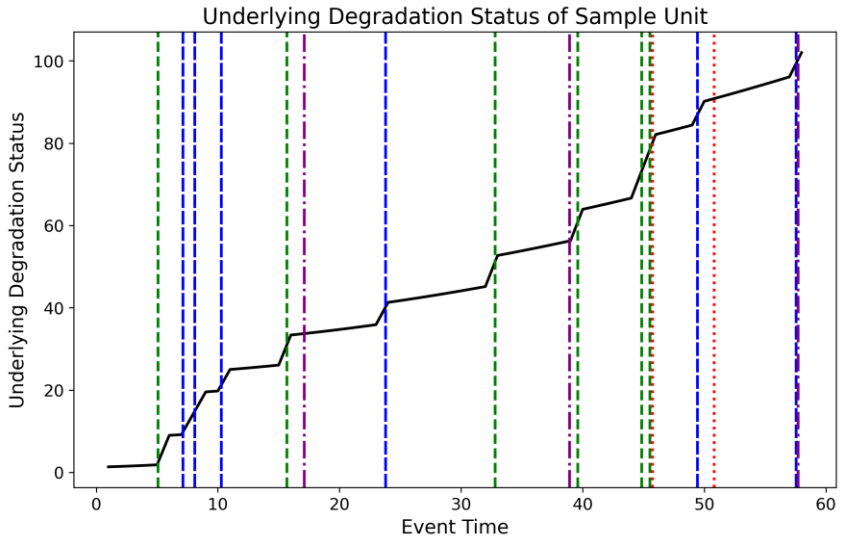


Figure 5.5 Underlying degradation status of a sample training unit. (Blue Long Dashed: Event 1 occurrence time, Green Short Dashed: Event 2 occurrence time, Purple Dashdotted: Event 3 occurrence time, Orange Dotted: Event 4 occurrence time, Black Solid: Underlying degradation status)

5.4.1.2 Performance Evaluation & Benchmark Methods

This study thoroughly evaluates various aspects of the UINN model:

- First, the benefits of incorporating insights from both data types are assessed. The prognostic performance of the UINN model is compared with model variants that only consider information from a single data type. The three variants are the full UINN model that considers both signal and event data (i.e., "Full"), the UINN model that only considers event data (i.e., "Event only"), and the UINN model that only considers signal data (i.e., "Signal only").
- In addition, the event type/time prediction performance is compared with four popular deep learning and statistical benchmarks. These include: (i) the Hawkes process with a nonparametric kernel [122], (ii) the bi-directional LSTM and (iii) the GRU approach by Huang et al. [103], and (iv) the popular transformers architecture [117]. The first approach

using Hawkes process is a statistical method that performs a nonparametric estimation of the unknown kernel function using the expectation maximization algorithm. The remaining three benchmarks are DL methods that leverage various architectures commonly used in event type/time prediction. Both LSTM and GRUs are based on a recurrent neural network architecture, while the more modern transformers use attention in place of the recurrent relations.

- Finally, this study explores the benefits of considering uncertainty information in the joint training procedure. A naïve version of the UINN model without uncertainty information is compared with the UINN model with uncertainty information. The predictive performance and the training curves of the two models are further analyzed.

The models are evaluated on three metrics. For RUL prediction, signal prediction, and event time prediction, the mean absolute error (MAE) metric is used:

$$MAE_{RUL} = \frac{\sum_{i=1}^{N_{test}} \sum_{j=TW_{sig}+1}^{n_{i,signal}} \left| RUL_{i,j} - \widehat{RUL}_{i,j} \right|}{\sum_{i=1}^{N_{test}} \left(n_{i,signal} - TW_{sig} \right)}, MAE_Y \frac{\sum_{i=1}^{N_{test}} \sum_{j=TW_{sig}+1}^{n_{i,signal}} \left| Y_{i,j} - \hat{Y}_{i,j} \right|}{\sum_{i=1}^{N_{test}} \left(n_{i,signal} - TW_{sig} \right)},$$

$$MAE_m = \frac{\sum_{i=1}^{N_{test}} \sum_{j=TW_{event}+1}^{n_{i,event}} \left| m_{i,j} - \hat{m}_{i,j} \right|}{\sum_{i=1}^{N_{test}} \left(n_{i,signal} - TW_{event} \right)}.$$

For event type prediction, the following micro F1 score and accuracy score are used:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N_{test}} \sum_{j=TW_{event}+1}^{n_{i,event}} \mathbb{I}\{e_{i,j} = \hat{e}_{i,j}\}}{\sum_{i=1}^{N_{test}} \left(n_{i,signal} - TW_{event} \right)}, \text{Micro F1}$$

$$= \frac{2 \times Micro\ Precision \times Micro\ Recall}{Micro\ Precision + Micro\ Recall},$$

$$\text{Micro Precision} = \frac{\sum_{z=1}^{Z} \text{TP}_\text{z}}{\sum_{z=1}^{Z} (\text{TP}_\text{z} + \text{FP}_\text{z})}, \text{Micro Recall} = \frac{\sum_{z=1}^{Z} \text{TP}_\text{z}}{\sum_{z=1}^{Z} (\text{TP}_\text{z} + \text{FN}_\text{z})},$$

where $\mathrm{TP_z} = \sum_{i=1}^{N_{test}} \sum_{j=TW_{event}+1}^{n_{i,event}} \mathbb{I}\{e_{i,j} = z \cap \hat{e}_{i,j} = z\}$, $\mathrm{FP_z} = \sum_{i=1}^{N_{test}} \sum_{j=TW_{event}+1}^{n_{i,event}} \mathbb{I}\{e_{i,j} \neq z \cap \hat{e}_{i,j} = z\}$, $\mathrm{FN_z} = \sum_{i=1}^{N_{test}} \sum_{j=TW_{event}+1}^{n_{i,event}} \mathbb{I}\{e_{i,j} = z \cap \hat{e}_{i,j} \neq z\}$, and $\mathbb{I}$ is the indicator function. Since there are multiple event types, the micro-averaged F1 score is employed here instead of the conventional F1 score for binary event types. Accuracy directly measures the percent of the correctly predicted event types, while the F1 score is a more balanced score that takes the harmonic mean of precision and recall.

5.4.1.3 Simulation Results

First, the event type prediction, signal prediction, and final RUL prediction results of the three model variants are shown in Table 5.1 below. Note that the numerical results are obtained by averaging the prediction errors across 50 repeated evaluations, and the lowest errors of each prediction task is boldfaced for visual clarity. Results show that the full UINN model that considers

Table 5.1 Evaluation results of the UINN model and the signal/event only counterparts with ± 1 standard deviation

Model	**Event Type (Micro F1)**	**Signal (MAE)**	**RUL (MAE)**
UINN (Full)	**0.9098±0.0275**	**0.1274±0.0365**	**7.7978±0.4410**
UINN (Event only)	0.9051±0.0290	20.474±0.0224	14.990±0.0580
UINN (Signal only)	0.2156±0.0096	0.1732±0.0653	7.9394±0.6281

both data types outperform the event/signal only counterparts. As expected, the models that only capture a single data type (i.e., signal/event only) shows poor performance in predicting the other data type as well as the final RUL values. Another interesting observation is that the full model has better predictive performance than individual models on their respective tasks (i.e., event type prediction and signal prediction). For instance, the full model has a marginally higher micro F1 score (0.9098) than the event only model's F1 score (0.9051), and a lower MAE (0.1274) than the

signal only model (0.1732). This further demonstrates that jointly modeling the dynamics of both data types and their associated prediction tasks improves the model's representation capacity, leading to improved performance over individual prediction tasks.

Table 5.2 Event prediction results of the UINN model against existing benchmark methods with ± 1 standard deviation

Model	**Event Type (Micro F1)**	**Event Type (Accuracy)**	**Event Time (MAE)**
Hawkes Process (nonparametric kernel)	0.6875±0.1289	0.6759±0.1372	1.1326±0.2314
LSTM [103]	0.8872±0.0344	0.8881±0.0305	0.6788±0.0881
GRU [103]	0.8936±0.0127	0.8946±0.0109	0.7152±0.0512
Transformers [117]	0.3487±0.0111	0.3474±0.0122	2.2829±0.0054
UINN (Full)	**0.9098±0.0275**	**0.9096±0.0279**	**0.6565±0.0465**

To further validate this result, Table 5.2 presents a detailed comparison of the event prediction performance of the UINN model against existing benchmark methods. Note that all benchmarks have been solely trained on the event data and have no access to the signal data. From the prediction results in Table 5.2, the UINN model outperforms all benchmarks in event type and time prediction. This again demonstrates the effectiveness of incorporating insights from both data types, as it results in superior prediction performance in individual tasks like event prediction. In particular, the methods by [103] perform similarly to the proposed model as they are based on a similar RNN architecture, but the proposed UINN model achieves a better performance in all categories. For the traditional Hawkes Process with a nonparametric kernel, it has overall much lower prediction performance due to its restrictive model structure and focus on parametricity. Modern methods like Transformers perform significantly worse than even the conventional statistical models. This is likely because these advanced models with attention often require a large amount of training data to effectively learn the underlying patterns. Additionally, these models tend to have many parameters that makes them prone to over/underfitting when data is limited.

Finally, this study investigates the contributions of the task-specific uncertainty information to the joint loss function. To demonstrate the benefits, this study first considers a naïve model with no uncertainty information, where the weights of each loss component are all equal such that w^{time}, w^{type}, $w^{sig}, w^{RUL} = 1$. This naïve model's training procedure and prognostic performance are compared to that of the proposed UINN model. Note that all models consider both event and signal data types, and the only difference is the inclusion of uncertainty in the loss function weights. Evaluation results in Table 5.3 show that the naïve model drastically underperforms the uncertainty-informed model in terms of event type, signal, and RUL prediction.

Table 5.3 Evaluation results of the UINN model with/without uncertainty information with ± 1 standard deviation

Model	**Event Type (Micro F1)**	**Signal (MAE)**	**RUL (MAE)**
UINN (With uncertainty)	**0.9098±0.0275**	**0.1274±0.0365**	**7.7978±0.4410**
UINN (Naïve)	0.2609±0.1748	0.2279±0.3652	7.9431±2.4100

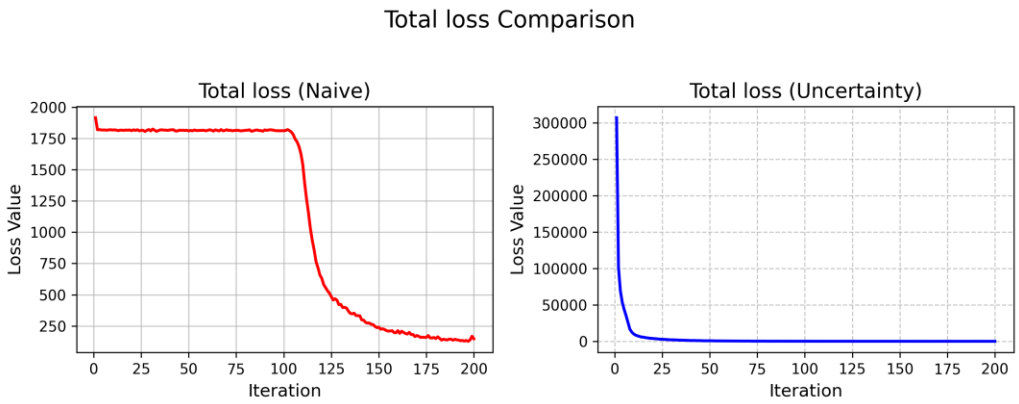


Figure 5.6 Averaged training total loss curves of the Naïve model in red (left) and the proposed UINN model with uncertainty-informed weights in blue (right).

Figure 5.6 illustrates the averaged total training loss of the naïve model and the proposed UINN model with uncertainty-informed weights. Note that the reported training curves are averaged across 50 iterations with different initializations. Due to the different scales of the loss functions, this analysis focuses on the convergence trends of the loss functions rather than their absolute values. Results show that uncertainty-informed weights significantly accelerate the convergence rate of the network. The loss function of the UINN model in blue (right) stabilizes in around 25 iterations, while the naïvely weighted model in red (left) requires more than 175 iterations. This phenomenon can be seen across all the different loss components, including the event type and event time loss as shown in Figure 5.7. From the figure, the proposed model with uncertainty converges much quicker, while the naïve versions suffer for around 100 initial iterations.

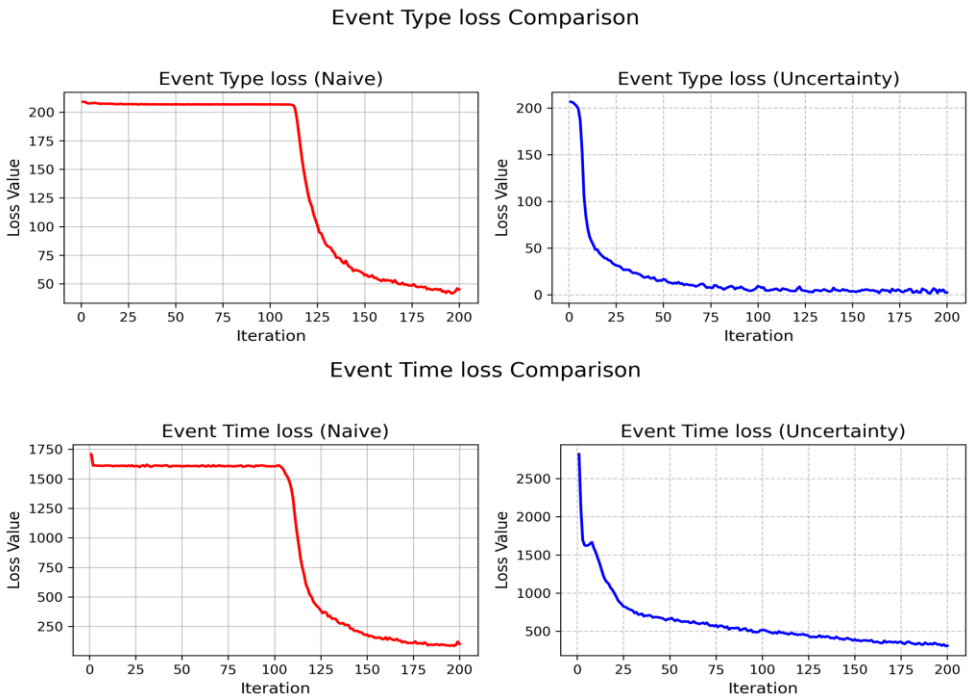


Figure 5.7 Visualized event type loss and event time loss training curves with the Naïve model in red (left) and the proposed UINN model with uncertainty-informed weights in blue (right).

These results further highlight the importance of leveraging uncertainty information in the loss function, as it results in faster model convergence and avoids potential fitting issues in the early

stages of training. Furthermore, using uncertainty information also leads to higher overall performance in event type, signal, and RUL prediction tasks.

5.4.2 Case Study

5.4.2.1 Experimental Setup & Data Collection

This real case study will further evaluate the proposed method using the battery status information from a PiSugar battery connected to a Raspberry Pi device. The Raspberry Pi is a

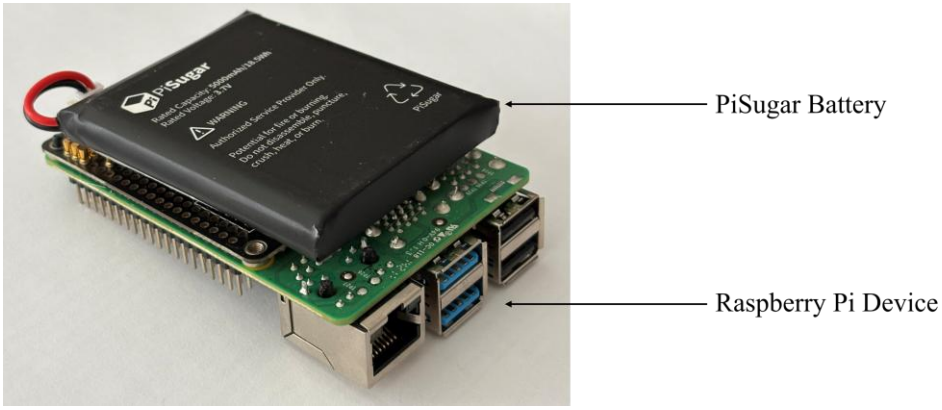


Figure 5.8 PiSugar 2 battery attached to a Raspberry Pi 4 Model B.

versatile, credit-card sized computer widely used for educational, research, and Internet of Things applications due to its affordability and ease of use. Typically, Raspberry Pi devices require a 15W USB-C cable power source to operate. Instead, one can attach a PiSugar portable battery on the Raspberry Pi, allowing it to operate in remote environments. For this case study, PiSugar 2 Lithium battery with a rated capacity of 5000mAh/18.5Wh and a rated voltage of 3.7V, and a Raspberry Pi 4 Model B were used. A picture of the PiSugar battery attached to the Raspberry Pi device is shown in Figure 5.8.

The PiSugar battery provides a battery level indicator that estimates and reports the remaining power level as a percentage. To emulate the different levels of initial wear and tear, each PiSugar battery starts with an initial power level between 95% and 100% and stops operation when the

battery level drops to 40%. The battery level indicator is collected every second and is treated as the system's continuous degradation signal. To introduce the effect of events, two types of artificial computational loads are imposed on the Raspberry Pi. The first event type (event 1) is a recursive Fibonacci computation task that brute-forces the Fibonacci numbers. This algorithm is very inefficient due to its exponential time complexity and puts a significant strain on the CPU resources. The second event type (event 2) is a sequential randomized matrix multiplication task, where the program successively generates large square matrices of dimension 2000 and continuously multiply them. Such large matrix multiplication tasks are known to consume significant CPU and memory resources. Initial experiments revealed that the recursive Fibonacci computation consumed on average 22% CPU and 15% memory, while the randomized matrix multiplication task consumed on average 44% CPU and 34% memory.

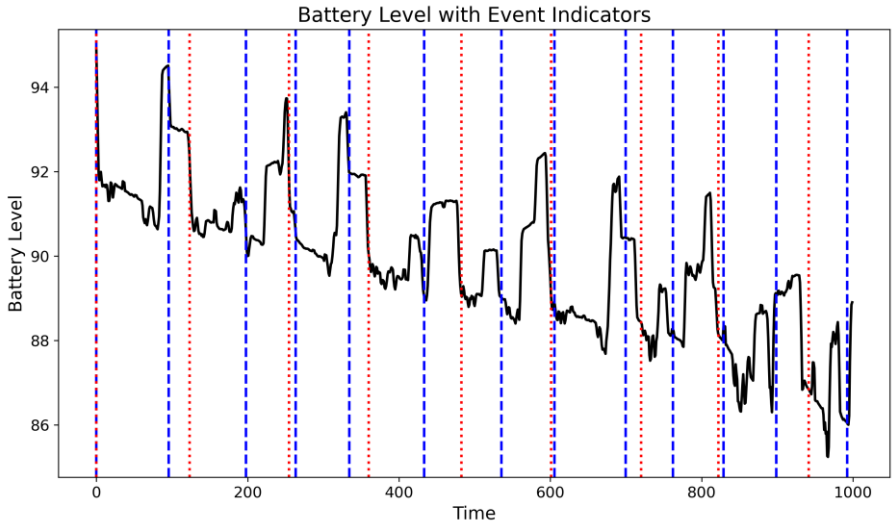


Figure 5.9 PiSugar battery level of a sample unit for the first 1000 seconds. (Black solid: battery level, Blue dashed: start time of event type 1, Red dotted: start time of event type 2)

Each event type is non-terminal and can occur multiple times throughout the unit's lifespan. Also, each event has an active period t_{active} and an inactive period $t_{inactive}$. The first event type (i.e., recursive Fibonacci) has an active period of t_{active}~Unif(30,60) and an inactive period of

$t_{inactive}$~Unif(10,20), while the second event type (i.e., randomized matrix multiplication) has an active period of t_{active}~Unif(60,90) and an inactive period of $t_{inactive}$~Unif(30,40). Note that both t_{active} and $t_{inactive}$ are in seconds.

The battery level of a sample unit is plotted in Figure 5.9, with the battery level shown in the black line, the start time of event 1 in blue, and the start time of event 2 in red. One interesting phenomenon is that the battery level sometimes tends to suddenly increase. This may seem counterintuitive as the battery is not charged during the experiment. However, this is an expected behaviour of the battery as under load, the battery's voltage drops due to internal resistance and increased current draw, leading the battery level indicators to show a lower charge. Once the load ends, the battery's voltage recovers slightly as it "bounces back" to its resting voltage. This recovery can cause the battery level indicator to show a higher charge level [123]. Overall, the dataset contains records of 15 units with on average 6000 signal observations.

5.4.2.2 Case Study Results

To demonstrate the benefits of considering both data types, the performance of the UINN model is evaluated against the event only model and the signal only model. The same evaluation metrics are used as the simulation study, with the micro F1 score for event type prediction, and the MAE for signal and RUL prediction. An 80/20 train/test split with 12 training units and 3 test units is repeated 10 times for a fair comparison. The evaluation results are available in Table 5.4, with the lowest average errors boldfaced for visual clarity.

Table 5.4 Evaluation results of the UINN model on the case study dataset

Model	**Event Type (Micro F1)**	**Signal (MAE)**	**RUL (MAE)**
UINN (Full)	**0.9764±0.0869**	**1.3674±0.2490**	**963.979±77.25**
UINN (Event only)	0.9676±0.0892	69.321±0.1982	1428.58±55.73
UINN (Signal only)	0.4403±0.0547	1.4690±0.2948	1333.94±66.54

Results from Table 5.4 align closely with the findings from the simulation study. First, the full UINN model demonstrates superior prediction performance by leveraging insights from both data types. Most importantly, it has a drastically better prognostic performance than its counterparts with a significantly lower RUL prediction error. Second, incorporating both data types also enhances the prediction performance of individual tasks such as event type prediction and signal prediction. For event type prediction, the full model has higher Micro F1 (0.9764) than the event predictor only model (0.9676). Similarly, the signal prediction error of the full model (1.3674) is lower than the signal predictor only model (1.4690). Hence, considering both data types and their associated tasks not only improves overall prognostic performance but also enhances the accuracy of individual prediction tasks.

5.5 Conclusion

This study proposed a novel uncertainty-informed neural network model for extracting prognostic insights from multi-type degradation-related data. Despite recent developments in prognostics, one key limitation of existing methods is the lack of a holistic prognostic model that can effectively accommodate both discrete event data and continuous signal data into the final RUL predictions. To overcome this issue, the proposed UINN model has two predictors that capture the unique dynamics of each data type and integrate them into the final RUL prediction. Then, all predictors in the UINN model are jointly trained to prevent introducing unwanted bias. One challenge of jointly training such a complex network with multiple predictors and data types is that it can easily encounter over/underfitting issues. This can result in sub-optimal model performance, as the model might not fully learn or capture the dynamics of a specific data type. To avoid such training issues, the UINN model leverages task-specific uncertainty as the weights for each loss component. These task-specific uncertainties are treated as learnable parameters, and

the model automatically assigns larger weights to tasks with lower uncertainties and smaller weights to tasks with higher uncertainties. The extensive numerical studies on simulated data as well as the case study data from PiSugar batteries demonstrate the superior performance of the UINN model relative to existing benchmark methods. Specifically, the UINN method not only achieved higher performance in prognostic tasks like RUL prediction, but also on next event type, time, and signal prediction. Furthermore, a close examination of the training curves showed that the uncertainty information avoided underfitting during model training.

There are several promising directions for future work. First, the current UINN framework could be extended to incorporate additional data modalities. The UINN framework is designed for two primary data types: discrete event data and continuous signal data. However, modern manufacturing systems often generate other data types, such as text-based maintenance or operational logs. Integrating these different data modalities could further improve the accuracy of RUL predictions. Second, the UINN model provides point estimates of the RUL instead of interval estimates. In degradation applications, it is recommended to have interval estimates of the RUL due to its inherent stochastic nature. Therefore, one can explore how to integrate the task-specific uncertainty to provide accurate uncertainty quantifications of the final RUL predictions.

5.6 Appendix

5.6.1 Average training time and hyperparameter optimization for simulation study

This section describes the hyperparameter optimization procedure for the simulation study of the proposed UINN model. Specifically, the hyperparameters were optimized using a two-stage approach. In the first stage, the event and signal predictors were individually optimized on their respective tasks (i.e., event type/time prediction and signal prediction) using a grid-based search.

This step ensures that each predictor accurately captures the dynamics of each data type. In the second stage, the hyperparameters of the signal and event predictors were fixed to their optimized values, and a separate grid-based search was conducted to fine-tune the hyperparameters of the final RUL predictor. The optimized hyperparameters for each signal, event, and RUL predictor is listed in Table 5.5 below.

Table 5.5 Optimized hyperparameters for the simulation study

Model	**Optimized Hyperparameters**
Signal Predictor	Hidden Layers: 3 Hidden Nodes: [64,64,16] Embedding Dimension: 20 Number of Attention Heads: 2 Learning Rate: 0.0001 Dropout Probability: 0.1 Batch Size: 32
Event Predictor	Hidden Layers: 2 Hidden Nodes: [64,64] Learning Rate: 0.0001 Dropout Probability: 0.1 Batch Size:50
RUL Predictor	Hidden Layers: 2 Hidden Nodes: [50,50] Learning Rate: 0.0001 Dropout Probability: 0.1 Batch Size: 16

Under this configuration, the proposed UINN model was trained for a maximum of 250 epochs with early stopping. Over the course of 10 iterations, the UINN model took an average of 71.253 minutes to finish training and took less than 250 epochs to converge. In contrast, the naïve variant (i.e., model trained without uncertainty information in the joint loss function shown in Table 5.3) exhausted the full 250 epochs and yielded poorer predictive performance. Hence, the proposed UINN model trains faster than the naïve model that do not leverage uncertainty information.

Chapter 6 Summary

Prognostics and degradation modeling are essential to ensure reliable performance of smart and connected systems. This dissertation addresses key challenges in data-driven degradation modeling and prognostics, with concentrations on improving explainability and alignment with existing domain knowledge. The key contributions of this dissertation can be organized as such.

Chapter 2 proposed a novel data-driven approach for modeling and predicting the progression of void swelling. The proposed model integrated nuclear engineering-specific domain knowledge such as shape constraints and the impact of covariates to accurately capture the behavior of void swelling processes. Due to the careful alignment with prior knowledge, the proposed model boasts superior predictive performance and produces nuclear physics-compliant results.

Chapter 3 introduced an integration uncertainty quantification (IUQ) model to capture the uncertainties from jointly modeling time-to-event data and longitudinal data. The proposed model produced accurate uncertainty quantifications by propagating the uncertainties from both data types. As a result, the IUQ model provided more reliable and calibrated uncertainty estimates and RUL predictions than existing approaches.

Chapter 4 then presented a Bayesian sensor selection algorithm for high-dimensional engineering systems. By leveraging a spike-and-slab prior on the sensors, the proposed method effectively identified informative sensors even under the presence of sensor correlation. The selected sensors can then be used by practitioners to gain more interpretable insights on the system dynamics.

Chapter 5 presented a deep learning framework for jointly extracting prognostic insights from discrete event data and continuous sensor signals. Compared to traditional models that rely on

either data type to extract prognostic insights, the proposed framework can effectively leverage both data types and obtain more accurate RUL predictions.

Chapter 7 References

[1] C. J. Lu and W. Q. Meeker, "Using Degradation Measures to Estimate a Time-to-Failure Distribution," *Technometrics*, vol. 35, no. 2, pp. 161–174, 1993, doi: 10.2307/1269661.

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

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

[4] M. Kim, C. Song, and K. Liu, "A Generic Health Index Approach for Multisensor Degradation Modeling and Sensor Selection," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 3, pp. 1426–1437, 2019, doi: 10.1109/TASE.2018.2890608.

[5] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process Mag*, vol. 30, no. 3, pp. 83–98, 2013.

[6] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[7] K. Liu and J. Shi, "IoT-Enabled System Informatics for Service Decision Making," *IEEE Intell Syst*, vol. 30, no. 6, pp. 18–21, 2015.

[8] C. S. Gray and S. J. Watson, "Physics of failure approach to wind turbine condition based maintenance," *Wind Energy*, vol. 13, no. 5, pp. 395–405, 2010.

[9] C. H. Oppenheimer and K. A. Loparo, "Physically based diagnosis and prognosis of cracked rotor shafts," in *Component and systems diagnostics, prognostics, and health management II*, SPIE, 2002, pp. 122–132.

[10] 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. 0, no. 0, p. 000, 2020, doi: 10.1080/24725854.2020.1766729.

[11] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *2017 IEEE international conference on prognostics and health management (ICPHM)*, IEEE, 2017, pp. 88–95.

[12] S. Zhong, S. Fu, L. Lin, X. Fu, Z. Cui, and R. Wang, "A novel unsupervised anomaly detection for gas turbine using isolation forest," in *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, IEEE, 2019, pp. 1–6.

[13] J. L. Straalsund, R. W. Powell, and B. A. Chin, "An overview of neutron irradiation effects in LMFBR materials," *Journal of Nuclear Materials*, vol. 108–109, pp. 299–305, 1982, doi: https://doi.org/10.1016/0022-3115(82)90499-8.

[14] C. Sun, F. A. Garner, L. Shao, X. Zhang, and S. A. Maloy, "Influence of injected interstitials on the void swelling in two structural variants of 304L stainless steel induced by self-ion

irradiation at 500 °C," *Nucl Instrum Methods Phys Res B*, vol. 409, pp. 323–327, Oct. 2017, doi: 10.1016/j.nimb.2017.03.070.

[15] F. A. Garner, "Irradiation performance of cladding and structural steels in liquid metal reactors," *Materials Science and Technology: A Comprehensive Treatment*, vol. 10, pp. 419–543, Oct. 1994.

[16] M. Jin, P. Cao, and M. P. Short, "Predicting the onset of void swelling in irradiated metals with machine learning," *Journal of Nuclear Materials*, vol. 523, pp. 189–197, Sep. 2019, doi: 10.1016/j.jnucmat.2019.05.054.

[17] A. S. Kalchenko, V. v. Bryk, N. P. Lazarev, I. M. Neklyudov, V. N. Voyevodin, and F. A. Garner, "Prediction of swelling of 18Cr10NiTi austenitic steel over a wide range of displacement rates," *Journal of Nuclear Materials*, vol. 399, no. 1, pp. 114–121, Apr. 2010, doi: 10.1016/j.jnucmat.2010.01.010.

[18] T. ni Yang *et al.*, "Influence of irradiation temperature on void swelling in NiCoFeCrMn and NiCoFeCrPd," *Scr Mater*, vol. 158, pp. 57–61, Jan. 2019, doi: 10.1016/j.scriptamat.2018.08.021.

[19] E. Wakai, N. Hashimoto, J. P. Robertson, T. Sawai, and A. Hishinuma, "Swelling of cold-worked austenitic stainless steels irradiated in HFIR under spectrally tailored conditions," *Journal of Nuclear Materials*, vol. 307–311, pp. 352–356, 2002, doi: https://doi.org/10.1016/S0022-3115(02)01189-3.

[20] J. A. Hudson, "Void formation in solution-treated aisi 316 and 321 stainless steels under 46.5 mev ni6+ irradiation," *Journal of Nuclear Materials*, vol. 60, no. 1, pp. 89–106, 1976, doi: https://doi.org/10.1016/0022-3115(76)90121-5.

[21] R. J. M. Konings and R. Stoller, *Comprehensive nuclear materials*. Elsevier, 2020.

[22] Y. Li, S. Hu, X. Sun, F. Gao, C. H. Henager, and M. Khaleel, "Phase-field modeling of void evolution and swelling in materials under irradiation," in *Science China: Physics, Mechanics and Astronomy*, May 2011, pp. 856–865. doi: 10.1007/s11433-011-4316-y.

[23] A. Gelman, "Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper)," *Bayesian Anal*, vol. 1, no. 3, pp. 515–534, 2006.

[24] L. C. E. Huberts, M. Schoonhoven, and R. J. M. M. Does, "Multilevel process monitoring: A case study to predict student success or failure," *Journal of Quality Technology*, vol. 54, no. 2, pp. 127–143, 2022.

[25] S. Watanabe, "A widely applicable Bayesian information criterion," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 867–897, 2013.

[26] R. Stiratelli, N. Laird, and J. H. Ware, "Random-Effects Models for Serial Observations with Binary Response," *Biometrics*, vol. 40, no. 4, pp. 961–971, 1984, doi: 10.2307/2531147.

[27] W. J. Browne and D. Draper, "A comparison of Bayesian and likelihood-based methods for fitting multilevel models," *Bayesian Anal*, vol. 1, no. 3, pp. 473–514, 2006.

[28] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[29] H. Joe, “Generating random correlation matrices based on partial correlations,” *J Multivar Anal*, vol. 97, no. 10, pp. 2177–2189, 2006.

[30] Stan Development Team, “Stan Modeling Language Users Guide and Reference Manual, Version 2.30.” [Online]. Available: https://mc-stan.org

[31] M. D. Hoffman and A. Gelman, “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo,” 2014. [Online]. Available: http://mcmc-jags.sourceforge.net

[32] T. G. Dietterich, “Ensemble methods in machine learning,” in *International workshop on multiple classifier systems*, Springer, 2000, pp. 1–15.

[33] L. Breiman, “Random forests,” *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001.

[34] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Ann Stat*, pp. 1189–1232, 2001.

[35] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[36] E. V Bonilla, K. Chai, and C. Williams, “Multi-task Gaussian process prediction,” *Adv Neural Inf Process Syst*, vol. 20, 2007.

[37] P.-C. Bürkner, “brms: An R Package for Bayesian Multilevel Models Using Stan,” *J Stat Softw*, vol. 80, no. 1, pp. 1–28, Aug. 2017, doi: 10.18637/jss.v080.i01.

[38] N. Igata, Y. Kohno, N. Tanabe, F. Rotman, and H. Tsunakawa, “Effects of nitrogen and carbon on void swelling in austenitic stainless steels,” *Journal of Nuclear Materials*, vol. 122, no. 1–3, pp. 219–223, May 1984, doi: 10.1016/0022-3115(84)90599-3.

[39] 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, 2018.

[40] Q. Zhou, J. Son, S. Zhou, X. Mao, and M. Salman, “Remaining useful life prediction of individual units subject to hard failure,” *IIE Transactions (Institute of Industrial Engineers)*, vol. 46, no. 10, pp. 1017–1030, Oct. 2014, doi: 10.1080/0740817X.2013.876126.

[41] J. Son, Q. Zhou, S. Zhou, X. Mao, and M. Salman, “Evaluation and comparison of mixed effects model based prognosis for hard failure,” *IEEE Trans Reliab*, vol. 62, no. 2, pp. 379–394, 2013, doi: 10.1109/TR.2013.2259205.

[42] Z. Li, S. Zhou, S. Choubey, and C. Sievenpiper, “Failure event prediction using the Cox proportional hazard model driven by frequent failure signatures,” *IIE Transactions (Institute of Industrial Engineers)*, vol. 39, no. 3, pp. 303–315, 2007, doi: 10.1080/07408170600847168.

[43] H. Liao, W. Zhao, and H. Guo, “Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model,” *Proceedings - Annual Reliability and Maintainability Symposium*, pp. 127–132, 2006, doi: 10.1109/RAMS.2006.1677362.

[44] X. Yue and R. Al Kontar, “Joint Models for Event Prediction From Time Series and Survival Data,” *Technometrics*, vol. 63, no. 4, pp. 477–486, 2021, doi: 10.1080/00401706.2020.1832582.

[45] R. R. Zhou, N. Serban, and N. Gebraeel, "DEGRADATION MODELING APPLIED TO RESIDUAL LIFETIME PREDICTION USING FUNCTIONAL DATA ANALYSIS," *Ann Appl Stat*, vol. 5, no. 2B, pp. 1586–1610, 2011, [Online]. Available: http://www.jstor.org/stable/23024864

[46] A. de Rose and A. Pallara, "Survival Trees: An Alternative Non-Parametric Multivariate Technique for Life History Analysis," *Eur J Popul*, vol. 13, no. 3, pp. 223–241, 1997, [Online]. Available: http://www.jstor.org/stable/20164002

[47] L.-J. Wei, "The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis," *Stat Med*, vol. 11, no. 14-15, pp. 1871–1879, 1992.

[48] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network," *BMC Med Res Methodol*, vol. 18, no. 1, p. 24, 2018, doi: 10.1186/s12874-018-0482-1.

[49] H. Kvamme, Ø. Borgan, and I. Scheel, "Time-to-Event Prediction with Neural Networks and Cox Regression," 2019. [Online]. Available: http://jmlr.org/papers/v20/18-424.html.

[50] P. Ferdinand Christ *et al.*, "SurvivalNet: Predicting patient survival from diffusion weighted magnetic resonance images using cascaded fully convolutional and 3D convolutional neural networks," *Proceedings - International Symposium on Biomedical Imaging*, pp. 839–843, Feb. 2017, doi: 10.48550/arxiv.1702.05941.

[51] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[52] W. Wang and A. H. Christer, "Towards a General Condition Based Maintenance Model for a Stochastic Dynamic System," *J Oper Res Soc*, vol. 51, no. 2, p. 145, Feb. 2000, doi: 10.2307/254254.

[53] Y. Wen, X. Guo, J. Son, and J. Wu, "A neural-network-based proportional hazard model for IoT signal fusion and failure prediction," *IISE Trans*, pp. 1–15, Jan. 2022, doi: 10.1080/24725854.2022.2030881.

[54] J. G. Ibrahim, M.-H. Chen, and D. Sinha, "BAYESIAN METHODS FOR JOINT MODELING OF LONGITUDINAL AND SURVIVAL DATA WITH APPLICATIONS TO CANCER VACCINE TRIALS," *Stat Sin*, vol. 14, no. 3, pp. 863–883, 2004, [Online]. Available: http://www.jstor.org/stable/24307419

[55] J. L. Wang, J. M. Chiou, and H. G. Müller, "Functional Data Analysis," *https://doi.org/10.1146/annurev-statistics-041715-033624*, vol. 3, pp. 257–295, Jun. 2016, doi: 10.1146/ANNUREV-STATISTICS-041715-033624.

[56] D. Rizopoulos, "Dynamic Predictions and Prospective Accuracy in Joint Models for Longitudinal and Time-to-Event Data," *Biometrics*, vol. 67, no. 3, pp. 819–829, Sep. 2011, doi: 10.1111/J.1541-0420.2010.01546.X.

[57] J. Miguel Hernández-Lobato and R. P. Adams, "Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks".

[58] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*, PMLR, 2016, pp. 1050–1059.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014, [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[60] Y. Gal, “Uncertainty in deep learning,” 2016.

[61] A. A. Tsiatis and M. Davidian, “Joint modeling of longitudinal and time-to-event data: an overview,” *Stat Sin*, pp. 809–834, 2004.

[62] A. A. Tsiatis, V. Degruttola, and M. S. Wulfsohn, “Modeling the relationship of survival to longitudinal data measured with error. Applications to survival and cd4 counts in patients with aids,” *J Am Stat Assoc*, vol. 90, no. 429, pp. 27–37, 1995, doi: 10.1080/01621459.1995.10476485.

[63] F. Yao, H.-G. Müller, and J.-L. Wang, “Functional data analysis for sparse longitudinal data,” *J Am Stat Assoc*, vol. 100, no. 470, pp. 577–590, 2005.

[64] J. T. Johns, C. Crainiceanu, V. Zipunnikov, and J. Gellar, “Variable-Domain Functional Principal Component Analysis,” *Journal of Computational and Graphical Statistics*, vol. 28, no. 4, pp. 993–1006, Oct. 2019, doi: 10.1080/10618600.2019.1604373.

[65] B. Efron, “Logistic regression, survival analysis, and the Kaplan-Meier curve,” *J Am Stat Assoc*, vol. 83, no. 402, pp. 414–425, 1988.

[66] N. Breslow, “Covariance analysis of censored survival data,” *Biometrics*, pp. 89–99, 1974.

[67] M. Campolieti, “Bayesian Estimation and Smoothing of the Baseline Hazard in Discrete Time Duration Models,” vol. 82, no. 4, pp. 685–694, 2000, Accessed: Feb. 15, 2023. [Online]. Available: https://www.jstor.org/stable/2646662

[68] D. R. Cox, “Regression models and life-tables,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 34, no. 2, pp. 187–202, 1972.

[69] S. Zhou and Y. Chen, *Industrial Data Analytics for Diagnosis and Prognosis: A Random Effects Modelling Approach*. John Wiley & Sons, 2021.

[70] “J2801_201805: Comprehensive Life Test for 12 V Automotive Storage Batteries - SAE International.” Accessed: Feb. 27, 2023. [Online]. Available: https://www.sae.org/standards/content/j2801_201805/

[71] J.-M. Kim, N. Wang, and Y. Liu, “Multi-stage change point detection with copula conditional distribution with PCA and functional PCA,” *Mathematics*, vol. 8, no. 10, p. 1777, 2020.

[72] L. Feng, H. Wang, X. Si, and H. Zou, “A state-space-based prognostic model for hidden and age-dependent nonlinear degradation process,” *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 4, pp. 1072–1086, 2013.

[73] J. Yu, “State-of-health monitoring and prediction of lithium-ion battery using probabilistic indication and state-space model,” *IEEE Trans Instrum Meas*, vol. 64, no. 11, pp. 2937–2949, 2015.

[74] 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, 2018, doi: 10.1080/24725854.2018.1440673.

[75] K. Liu, N. Z. Gebraeel, and J. Shi, “A Data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis,” *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 652–664, 2013, doi: 10.1109/TASE.2013.2250282.

[76] C.-G. Huang, X. Yin, H.-Z. Huang, and Y.-F. Li, “An enhanced deep learning-based fusion prognostic method for RUL prediction,” *IEEE Trans Reliab*, vol. 69, no. 3, pp. 1097–1109, 2019.

[77] J. Deutsch and D. He, “Using deep learning-based approach to predict remaining useful life of rotating components,” *IEEE Trans Syst Man Cybern Syst*, vol. 48, no. 1, pp. 11–20, 2017.

[78] R. Badawy *et al.*, “Automated quality control for sensor based symptom measurement performed outside the lab,” *Sensors*, vol. 18, no. 4, p. 1215, 2018.

[79] J. Fan, F. Han, and H. Liu, “Challenges of big data analysis,” *Natl Sci Rev*, vol. 1, no. 2, pp. 293–314, 2014.

[80] J. Vetelino and A. Reghu, *Introduction to sensors*. CRC press, 2017.

[81] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4, no. 4. Springer, 2006.

[82] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[83] H. Zou, “The adaptive lasso and its oracle properties,” *J Am Stat Assoc*, vol. 101, no. 476, pp. 1418–1429, 2006.

[84] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *J Am Stat Assoc*, vol. 96, no. 456, pp. 1348–1360, 2001.

[85] C.-H. Zhang, “Nearly unbiased variable selection under minimax concave penalty,” 2010.

[86] X. Fang, K. Paynabar, and N. Gebraeel, “Multistream sensor fusion-based prognostics model for systems with single failure modes,” *Reliab Eng Syst Saf*, vol. 159, pp. 322–331, 2017.

[87] M. G. Tadesse and M. Vannucci, “Handbook of bayesian variable selection,” 2021.

[88] J. Yu, S. Li, X. Liu, Y. Gao, S. Wang, and C. Liu, “Dynamic convolutional gated recurrent unit attention auto-encoder for feature learning and fault detection in dynamic industrial processes,” *Int J Prod Res*, vol. 61, no. 21, pp. 7434–7452, 2023.

[89] M. Kim, J. R. C. Cheng, and K. Liu, “An adaptive sensor selection framework for multisensor prognostics,” *https://doi.org/10.1080/00224065.2021.1960934*, vol. 53, no. 5, pp. 566–585, 2021, doi: 10.1080/00224065.2021.1960934.

[90] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[91] K. Liu, N. Z. Gebraeel, and J. Shi, “A Data-Level Fusion Model for Developing Composite Health Indices for Degradation Modeling and Prognostic Analysis,” *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 652–664, 2013, doi: 10.1109/TASE.2013.2250282.

[92] 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, 2019, doi: 10.1080/24725854.2018.1555384.

[93] N. N. Narisetty and X. He, “Bayesian variable selection with shrinking and diffusing priors,” *The Annals of Statistics*, vol. 42, no. 2, pp. 789–817, Apr. 2014, doi: 10.1214/14-AOS1207.

[94] H. Ishwaran and J. S. Rao, “Spike and slab variable selection: frequentist and Bayesian strategies,” 2005.

[95] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans Pattern Anal Mach Intell*, no. 6, pp. 721–741, 1984.

[96] N. N. Narisetty, J. Shen, and X. He, “Skinny gibbs: A consistent and scalable gibbs sampler for model selection,” *J Am Stat Assoc*, 2018.

[97] M. M. Barbieri and J. O. Berger, “Optimal predictive model selection,” *The Annals of Statistics*, vol. 32, no. 3, pp. 870–897, Jun. 2004, doi: 10.1214/009053604000000238.

[98] B. Leimkuhler and S. Reich, *Simulating hamiltonian dynamics*, no. 14. Cambridge university press, 2004.

[99] P. Breheny and J. Huang, “Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection,” *Ann Appl Stat*, vol. 5, no. 1, p. 232, 2011.

[100] P. Carbonetto and M. Stephens, “Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies,” *Bayesian Anal*, vol. 7, no. 1, pp. 73–108, 2012, doi: 10.1214/12-BA703.

[101] N. Simon, J. Friedman, R. Tibshirani, and T. Hastie, “Regularization Paths for Cox’s Proportional Hazards Model via Coordinate Descent,” *J Stat Softw*, vol. 39, no. 5, pp. 1–13, 2011, doi: 10.18637/jss.v039.i05.

[102] D. K. Frederick, J. A. Decastro, and J. S. Litt, “User’s Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS),” 2007, Accessed: Apr. 10, 2024. [Online]. Available: http://www.sti.nasa.gov

[103] C. Huang, A. Deep, S. Zhou, and D. Veeramani, “A deep learning approach for predicting critical events using event logs,” *Qual Reliab Eng Int*, vol. 37, no. 5, pp. 2214–2234, 2021, doi: 10.1002/qre.2853.

[104] Q. Zhou, J. Son, S. Zhou, X. Mao, and M. Salman, “Remaining useful life prediction of individual units subject to hard failure,” *IIE Transactions*, vol. 46, no. 10, pp. 1017–1030, 2014.

[105] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mech Syst Signal Process*, vol. 107, pp. 241–265, 2018.

[106] M. W. Arisido, L. Antolini, D. P. Bernasconi, M. G. Valsecchi, and P. Rebora, "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 Med Res Methodol*, vol. 19, no. 1, p. 222, 2019, doi: 10.1186/s12874-019-0873-y.

[107] J. Dupuy and M. Mesbah, "Joint modeling of event time and nonignorable missing longitudinal data," *Lifetime Data Anal*, vol. 8, pp. 99–115, 2002.

[108] E. A. Pena, R. L. Strawderman, and M. Hollander, "Nonparametric estimation with recurrent event data," *J Am Stat Assoc*, vol. 96, no. 456, pp. 1299–1315, 2001.

[109] L. Doyen, O. Gaudoin, and A. Syamsundar, "On geometric reduction of age or intensity models for imperfect maintenance," *Reliab Eng Syst Saf*, vol. 168, pp. 40–52, 2017.

[110] A. Deep, D. Veeramani, and S. Zhou, "Event Prediction for Individual Unit Based on Recurrent Event Data Collected in Teleservice Systems," *IEEE Trans Reliab*, vol. 69, no. 1, pp. 216–227, 2020, doi: 10.1109/TR.2019.2909471.

[111] O. Shchur, A. C. Türkmen, T. Januschowski, and S. Günnemann, "Neural temporal point processes: A review," *arXiv preprint arXiv:2104.03528*, 2021.

[112] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007.

[113] P. F. Christ *et al.*, "SurvivalNet: Predicting patient survival from diffusion weighted magnetic resonance images using cascaded fully convolutional and 3D convolutional neural networks," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, 2017, pp. 839–843.

[114] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1555–1564.

[115] T. Omi, K. Aihara, and others, "Fully neural network based model for general temporal point processes," *Adv Neural Inf Process Syst*, vol. 32, 2019.

[116] S. Xiao, J. Yan, X. Yang, H. Zha, and S. Chu, "Modeling the intensity function of point process via recurrent neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[117] A. Vaswani, "Attention is all you need," *Adv Neural Inf Process Syst*, 2017.

[118] J. D. Singer and J. B. Willett, "It's about time: Using discrete-time survival analysis to study duration and the timing of events," *Journal of educational statistics*, vol. 18, no. 2, pp. 155–195, 1993.

[119] R. Caruana, "Multitask learning," *Mach Learn*, vol. 28, pp. 41–75, 1997.

[120] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," *Adv Neural Inf Process Syst*, vol. 31, 2018.

[121] E. Bacry, M. Bompaire, S. Gaïffas, and S. Poulsen, “Tick: a Python library for statistical learning, with a particular emphasis on time-dependent modelling,” 2018. [Online]. Available: https://arxiv.org/abs/1707.03003

[122] E. Lewis and G. Mohler, “A nonparametric EM algorithm for multiscale Hawkes processes,” *J Nonparametr Stat*, vol. 1, no. 1, pp. 1–20, 2011.

[123] J. F. Manwell and J. G. McGowan, “Lead acid battery storage model for hybrid energy systems,” *Solar energy*, vol. 50, no. 5, pp. 399–405, 1993.