Analysis of Large Heterogeneous Repairable System Reliability Data with Static System Attributes and Dynamic Sensor Measurement in Big Data Environment

Xiao Liu[1] and Rong Pan[2]

[1]Department of Industrial Engineering, University of Arkansas
[2]School of Computing, Informatics, Decision Systems Engineering, Arizona State University

Abstract

In the age of Big Data, one pressing challenge facing engineers is to perform reliability analysis for a large fleet of heterogeneous repairable systems with covariates. In addition to static covariates, which include time-invariant system attributes such as nominal operating conditions, geo-locations, etc., the recent advances of sensing technologies have also made it possible to obtain dynamic sensor measurement of system operating and environmental conditions. As a common practice in the Big Data environment, the massive reliability data are typically stored in some distributed storage systems. Leveraging the power of modern statistical learning, this paper investigates a statistical approach which integrates the Random Forests algorithm and the classical data analysis methodologies for repairable system reliability, such as the nonparametric estimator for the Mean Cumulative Function and the parametric models based on the Nonhomogeneous Poisson Process. We show that the proposed approach effectively addresses some common challenges arising from practice, including system heterogeneity, covariate selection, model specification and data locality due to the distributed data storage. The large sample properties as well as the uniform consistency of the proposed estimator is established. Two numerical examples and a case study are presented to illustrate the application of the proposed approach. The strengths of the proposed approach are demonstrated by comparison studies.

Key words: *Repairable Reliability Data Analysis, Recurrence Data, Random Forests, Mean Cumulative Function, Nonhomogeneous Poisson Process.*

1 Introduction

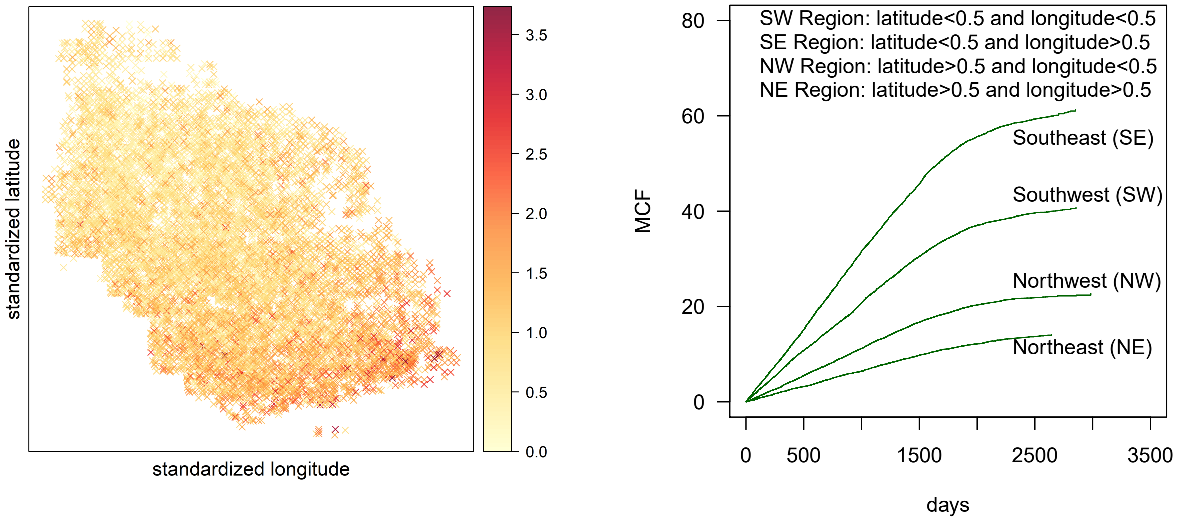
Advances in sensing technologies are re-shaping the landscape of statistical analysis for reliability data. For example, a typical repairable system reliability data set contains the failure times from a large fleet of systems as well as a set of covariates associated with each system. Some covariates are static such as time-invariant system attributes, while some covariates are dynamic such as sensor measurements of operating and environmental conditions (known as the SOE data). SOE data represent one of the most significant trends in modern reliability analysis in the age of Big Data (Meeker and Hong, 2014; Hong et al., 2018). To elaborate the challenges lie ahead, a motivating example is firstly presented.

1.1 Motivating Example and Challenges

One million oil and natural gas wells are currently operating in the U.S. (USEIA, 2017). Preventive maintenance of these complex engineering systems is managed by multinational oil and gas service companies which collect and store massive field data sets. Our motivating example is based on a data set which consists of the failure/repair data, system attributes, and sensor measurement of 8232 oil and gas wells installed between 2007 and 2017.

For each well, the ages upon failures are recorded. Figure 1(a) shows the geo-locations and the average annual number of failures of these wells. The longitude and latitude are standardized to a unit square $[0, 1]^2$ so as to keep the actual well locations confidential. Figure 1(b) shows the estimated MCF (Mean Cumulative Function) (Nelson, 1995) for systems over four geo-regions: southwest (SW), southeast (SE), northwest (NW) and northeast (NE). Systems in the southern area appear to have higher overall failure rates than those located in the central and northern regions. Such a spatial trend is often due to system age, soil type, environmental and operating conditions, and others.

In addition to the failure ages, a number of 15 covariates are available including eight *static* covariates, $x_1 \sim x_8$, and seven *dynamic* covariates, $z_1 \sim z_7$ as follows: 1) (well attributes) covariates $x_1 \sim x_6$ are basic well attributes such as well size, nominal working



(b)

Figure 1: Panel (a) shows the locations of the 8232 wells and the legend indicates the average annual failures; Panel (b) shows the estimated MCF over four geo-regions.

conditions, etc.; 2) (geographical location) covariates x_7 and x_8 respectively specify the latitude and longitude of each well. Neighboring wells share common environmental conditions such as temperature-humidity variation, soil type, contamination, etc. Although these factors are not directly observed, they may lead to some important spatial patterns of the failure processes; 3) (torque and load) covariates $z_1 \sim z_5$ are related to various dynamic torque and load of each well such as gear torque, load range, etc.; and 4) (stress) covariates z_6 and z_7 are respectively the gearbox and structural stress of each well monitored by sensors.

To analyze such a repairable system reliability data set, nonparametric graphical methods have been widely used for estimating the MCF which describes the average number of failures for a population of systems up to a certain age (Nelson, 1995; Meeker and Escobar, 1998). However, nonparametric graphical methods are less effective in modeling the complex relationship between failure processes and covariates. Parametric point and counting processes have also been widely used to model the recurrence data (Fleming and Harrington, 1991; Anderson et al., 1993; Meeker and Escobar, 1998; Rigdon and Basu, 2000). Commonly used models include the Nonhomogeneous Poisson Process (NHPP), renewal process, trend-renewal process, piecewise exponential model, bounded intensity model, etc. (Pulcini, 2001; Lindqvist et al., 2003; Pan and Rigdon, 2009; Yang, et al., 2012; Ye, et al., 2013b; Mittman, et al., 2018). However, as both the system fleet size and the number of covariates increase, some fundamental challenges arise in the modern Big Data analytics environment:

- (*Heterogeneity*) It is no longer appropriate to ignore system heterogeneity among a large fleet of field systems (Lindqvist et al., 2003; Stocker and Pena, 2007; Ye, et al., 2013; Xu, et al., 2017). In the motivating example, it is impossible to assume that the 8232 wells are from a homogeneous population. Subpopulations always exist due to a number of reasons including the basic system attributes, variation in operating and environmental conditions, maintenance history, etc. Although covariate information may be useful in explaining the system-to-system variation, multiple subpopulations can be governed by fundamentally different failure processes with distinctive dependence structure on covariates.

- (*Model Specification*) It is challenging to force and validate parametric assumptions that adequately describe the link between failure processes and covariates, especially when interactions and nonlinear effects exist among a large number of covariates. This issue is further complicated due to system heterogeneity: both the effects of individual covariates and the relationship between failure processes and covariates can vary across subpopulations.

- (*Model Complexity and Interpretability*) Although an increasingly larger number of covariates has been made available due to the advances of sensing technologies, it is evident from our industry practice that many covariates are in fact redundant from either the statistical modeling or domain knowledge perspective. However, as many companies today have invested tremendous resources in collecting and storing data, a common misconception arising from industry is that a statistical model needs to utilize all data provided. Hence, efficient selection of important covariates are particularly important in the Big Data environment, as data-driving models are becoming more complex but seemingly less interpretable.

- (*Data Locality*) In the Big Data environment, the ways data are stored determines how statistical data analysis can be efficiently performed—a critical issue whose impact on reliability analysis has often been overlooked. A common practice in industry today is to store the massive reliability data on distributed storage such as the Hadoop distributed file system. This leads to a critical difference between traditional statistical analysis and statistical analysis in the age of Big Data: traditional data analysis moves *data to algorithms*, while data analysis in the age of Big Data moves *algorithms to data*; as shown in Figure 2.

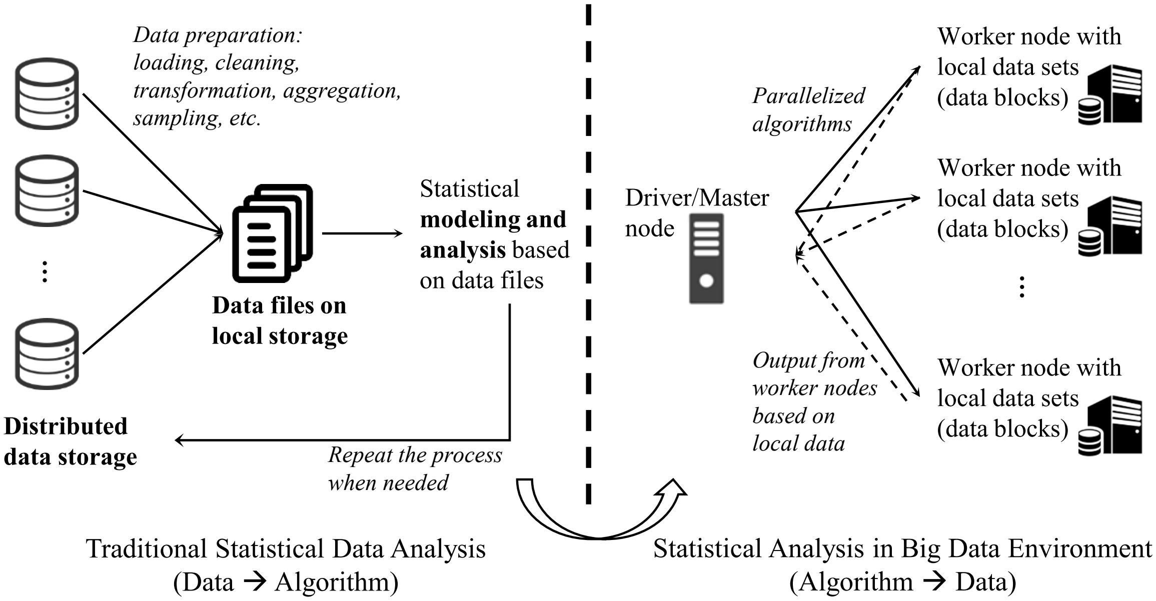


Figure 2: Traditional data analysis moves data to algorithms, while data analysis in the age of Big Data moves algorithms to data.

To elaborate, traditional data analysis often starts with data preparation which pulls data from a distributed storage with multiple servers. Then, through a series of data preprocessing, data files (such as text files and Excel spreadsheet) are generated and saved to a local storage. Only after such a time-consuming process can the modeling and statistical data analysis be performed based on the data files generated. Whenever there is an update on raw data sets in the distributed storage, or, there are new requirements arising from the modeling point of view, the above lengthy data preparation process is repeated which typically involves the transfer of a huge amount of data. Data analysis in the age of Big

Data works in exactly the opposite direction. Instead of pulling data from different servers, the statistical analysis procedures are assigned to different servers where data are stored in a distributed manner. This is known as "*Data Locality*" which has tremendous advantages in terms of reliability, scalability, efficiency and security (Apache Spark, 2018). The well-known MapReduce programming model—originally referred to the proprietary Google technology—is based on such an idea that any algorithms can be executed on smaller subsets of a larger data set, and the outputs generated from smaller subsets are then merged to form the final results. Through the idea of MapReduce, computations are parallelized on local nodes which is critical when working with large data.

1.2 Overview and Literature Review

To tackle the challenges discussed above, this paper investigates an approach which integrates two powerful methods respectively from the modern statistical learning and classical repairable system reliability analysis, i.e., the Random Forests (RF) algorithm (Breiman, 2001) and the nonparametric MCF estimator (Nelson, 1995). The RF method, which constructs ensembles (forests) from de-correlated base learners (trees), is one of the most popular ensemble methods which has gained tremendous success in practice. The randomness is introduced into the RF through two mechanisms: 1) each tree is grown based on a bootstrap sample (bagging), and 2) at each node of a tree, the optimum splitting variable and split point are chosen from a randomly sampled candidate variable set. The nonparametric MCF estimator, on the other hand, is one of the most successful methods for analyzing repairable system reliability data (i.e., recurrence data) in industry (Nelson, 1995; Meeker and Escobar, 1998; Doganaksoy et al., 2006; Zuo, et al., 2012). The theory behind the nonparametric MCF estimator is deeply rooted in point processes (Fleming and Harrington, 1991).

In the literature, tree-based methods for analyzing survival data (i.e., time-to-event data) with censoring have attracted much attention. A sum-of-trees (ensemble) model is essentially an additive model with multivariate components that effectively handle the complex interaction effects among covariates (Chipman et al., 2010; Pratola, et al., 2016). For right-censored data, Hothorn et al. (2006) introduced an RF algorithm and a generic gradient boosting algorithm for predicting the survival times of patients suffering from acute myeloid leukemia based on clinical and genetic covariates; also see Hothorn et al. (2004) for bagging survival trees. Ishwaran et al. (2008) proposed the Random Survival Forests (RSF) algorithm that integrates the classical Kaplan-Meier estimator into the framework of RF. Fan et al. (2006, 2009) investigated trees for correlated and multivariate survival data. Bacchetti and Segal (1995) proposed an approach which decomposes a subject into multiple pseudo-subjects, and the pseudo-subjects can be split across many nodes as a function of time. Bou-Hamad et al. (2009) incorporated dynamic covariates in the tree-based survival analysis by allowing subjects to be split across different nodes depending on the time windows. A review of tree-based methods for analyzing time-to-failure data can be found in Bou-Hamad (2011).

This work focuses on the analysis of large recurrence data with both static and dynamic covariates arising from repairable systems. We investigate the statistical properties of the estimator and show how the proposed approach naturally fits into the modern Big Data environment. To our best knowledge, such work has not yet been done in the literature. We organize the paper as follows: Section 2 presents the proposed algorithm by only considering static covariates. Technical details and some useful theoretical results (i.e., consistency and large-sample variance) are presented. In Section 3, we extend the framework to handle both static and dynamic covariates. Section 4 presents two numerical examples to demonstrate the advantages of the proposed method over some conventional methods. Section 5 revisits the motivating example and illustrates the application of the proposed approach.

2 The RF-R Algorithm

2.1 Overview

Consider a typical repairable system reliability data set which consists of n systems. For any system i ($i = 1, 2, ..., n$), we observe a vector $\boldsymbol{y}_i = (y_{i,1}, ..., y_{i,r_i}, c_i)$ which contains a sequence of r_i failure ages, $y_{i,1}, ..., y_{i,r_i}$, and the right censoring time c_i. Associated with system i there exists a number of p attributes, $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,p})$. Without loss of generality, it is assumed that all covariates are standardized on the unit interval $[0, 1]$, and the covariate space is hence a multi-dimensional cube $[0, 1]^p$. Let $\Lambda(t)$ denote the number of failures in the time interval $(0, t]$, then, the expectation of $\Lambda(t)$ is the MCF which characterizes the reliability of a population of repairable systems (Nelson, 1995; Meeker and Escobar, 1998).

Algorithm 1: The RF-R algorithm

```
Data: $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ for $i = 1, 2, ..., n$
Step 1 Draw $B$ bootstrap samples from the original data.
Step 2 for $b$=1,...,$B$ do
    Grow a random MCF tree by recursively repeating the following steps at each
     node of the tree:
    2.1) Select $m$ covariates at random from the $p$ covariates.
    2.2) Pick the best covariate and its split point among the $m$ selected covariates
     by maximizing the difference of the MCF between daughter nodes.
    2.3) Grow the tree to full size under the condition that each terminal node
     contains at least $d_0$ systems with at least one failure.
Step 3 Obtain the ensemble MCF by averaging the MCF obtained from all trees.
Step 4 Compute the Out-of-Bag (OOB) prediction error for the ensemble MCF.
```

To tackle the challenges described in Section 1.1, an algorithm, called RF-R (**R**andom **F**orests for **R**epairable System Reliability Analysis), is proposed to integrate the nonparametric MCF estimator into the framework of RF. The RF-R algorithm is described in Algorithm 1, while the technical and theoretical details are presented in Sections 2.2 and 2.3.

The RF-R integrates the nonparametric MCF estimator into the framework of RF, and hence has the same computational complexity as RF, i.e., $\mathcal{O}(mn \log n)$. The choice of m is case-dependent; for example, $m = \lfloor p/3 \rfloor$ as recommended in Hastie et al. (2009). By integrating the nonparametric MCF estimator with RF, the RF-R algorithm achieves the following advantages: 1) (heterogeneity) for each bootstrap sample, the algorithm uses a binary tree to divide the covariate space into a set of rectangular partitions represented by the terminal nodes of a tree. In other words, heterogeneous systems are divided into sub-groups based on their attributes. Then, separate reliability models, such as the MCF in this case, are constructed to characterize the system reliability on different terminal nodes; 2) (model specification) the nonparametric MCF estimator enables us to avoid the difficult specification of the complicated relationship between failure processes and covariates. Complex interaction structures in data can also be automatically captured by the tree-based methods (Chipman et al., 2010). The sample size in the Big Data environment is large enough to overcome the traditional limitations of nonparametric methods; and 3) (data locality) the RF-R algorithm, which can be implemented in a modern distributed computing environment, does not require any transfer of the original data sets across the worker nodes and satisfies the basic requirements for Big Data analytics as discussed in Section 1.1.

To elaborate, Step 1 involves generating bootstrap samples on local *worker* nodes. This process can be done in parallel across worker nodes. In Step 2, a MCF tree is grown which involves recursively splitting the tree nodes until the termination condition is met. For each splitting, the *driver* node selects m covariates at random from the p covariates, and creates the candidate split point for each of the selected m covariates. After receiving the selected covariates and the candidate split points, the *worker* node, in parallel, performs the splitting of the local bootstrap data into two sub-populations, and constructs the (local) MCF respectively for both sub-populations. The constructed MCF, using the data stored on the worker node, are returned to the *driver* node. After that, the *driver* node merges the local MCF constructed from all worker nodes using the method to be discussed in 2.2.1. Finally, the optimum splitting covariate and split-point are found on the *driver* node. The tree structure is updated on the *driver* node until the algorithm terminates.

2.2 Technical Details

Let $T(\boldsymbol{x};\Theta^{(b)})$ be a binary MCF tree based on the bootstrap sample b, $b=1,2,...,B$. Here, $\boldsymbol{x}$ is a vector of covariates and the parameter, $\Theta^{(b)}$, fully specifies the tree in terms of splitting covariates, split points, and the constructed MCF on terminal nodes. Let $(h_1^{(b)}, h_2^{(b)}, ..., h_M^{(b)})$ be a set of M terminal nodes of the bth tree and each terminal node defines a binary partition on the covariate space. Then, for a vector of covariates $\boldsymbol{x} \in h_j^{(b)}$, the MCF is given by:

$$\widehat{\text{MCF}}^{(b)}(t;\boldsymbol{x}) = \sum_{j=1}^{M} \widehat{\text{MCF}}_{h_j^{(b)}}(t) I\{\boldsymbol{x} \in h_j^{(b)}\} \tag{1}$$

where $\widehat{\text{MCF}}_{h_j^{(b)}}(t)$ is the MCF constructed on terminal node $h_j^{(b)}$. The bootstrap ensemble MCF is computed by averaging over the B trees:

$$\widehat{\text{MCF}}^{(*)}(t;\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} \widehat{\text{MCF}}^{(b)}(t;\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B}\sum_{j=1}^{M} \widehat{\text{MCF}}_{h_j^{(b)}}(t) I\{\boldsymbol{x} \in h_j^{(b)}\}. \tag{2}$$

The greedy algorithm can be used in step 2 of Algorithm 1 which recursively decides on the optimum splitting covariates, split points, and the tree topology (Hastie et al., 2009). Given the candidate splitting covariate j and the split point $\tilde{x}_j$, an intermediate node $\tilde{h}$ can be split into two daughter nodes. The greedy algorithm seeks the optimum splitting covariate and the split point that maximize the difference between the two MCF, $\widehat{\text{MCF}}_{\tilde{h}}^{(-)}(t)$ and $\widehat{\text{MCF}}_{\tilde{h}}^{(+)}(t)$, respectively constructed on the left and right daughter nodes:

$$\max_{j,\tilde{x}_j} g(\widehat{\text{MCF}}_{\tilde{h}}^{(-)}(t), \widehat{\text{MCF}}_{\tilde{h}}^{(+)}(t)) \tag{3}$$

where g is some distance measure between two MCF. Hence, two fundamental questions need to be addressed: the construction of MCF on a given tree node for distributed data (Section 2.2.1), and the choice of the splitting rule in (Section 2.2.2).

2.2.1 MCF for Large Distributed Data Sets

For distributed data sets, the MCF needs to be constructed in a distributed manner such that each worker node constructs the MCF using the local data on that worker node and the driver node merges the MCF constructed from all worker nodes. By modifying algorithm 16.1 in Meeker and Escobar (1998), we obtain the following algorithm which constructs MCF on a tree node using distributed data on worker nodes. This approach can be used to obtain the MCF on both intermediate and terminal nodes, and the subscript h in (3) is suppressed.

```
Algorithm 2: Nonparametric MCF for Distributed Data Sets
  Data: Prepare the data set $\mathfrak{D}_w$ on each local worker node $w$ ($w = 1, 2, ..., W$) from
        which a MCF is to be constructed.
  Step 1 (Map phase) For each worker node $w$, the following steps are executed in
   parallel, i.e., for $w$=1,...,$W$ do
    1.1. Find the ordered unique failure times, $t_1 < t_2 < ... < t_{n^{(w)}}$, where $n^{(w)}$ is the
         number of unique failure times in $\mathfrak{D}_w$.
    1.2. Compute the number of failure $d_i(t_k)$ for system $i$ at time $t_k$
         ($k = 1, 2, ..., n^{(w)}$) in $\mathfrak{D}_w$.
    1.3. Let $\delta_i(t_k) = 1$ if system $i$ is still being observed at $t_k$; $\delta_i(t_k) = 0$ otherwise.
    1.4. Compute the local MCF from the data set $\mathfrak{D}_w$ for $j = 1, 2, ..., n^{(w)}$

         $$\widehat{\mathrm{mcf}}^{(w)}(t_j) = \sum_{k=1}^{j} \left\{ \frac{\sum_{i=1}^{n^{(w)}} \delta_i(t) d_i(t_k)}{\sum_{i=1}^{n^{(w)}} \delta_i(t_k)} \right\} = \sum_{k=1}^{j} \frac{d_{\cdot}^{(w)}(t_k)}{\delta_{\cdot}^{(w)}(t_k)} = \sum_{k=1}^{j} \bar{d}^{(w)}(t_k) \qquad (4)$$

         where the estimate, $\widehat{\mathrm{mcf}}^{(w)}(t)$, is a step function, with jumps at failure times,
         but constant between the failure times.
  Step 2 (Reduce phase) Merge the local MCF on the driver node:

         $$\widehat{\mathrm{MCF}}(t) = \frac{\sum_{m=1}^{W} \widehat{\mathrm{mcf}}^{(w)}(t)}{W} \qquad (5)$$
```

From Nelson (1995) and Meeker and Escobar (1998), $\widehat{\mathrm{mcf}}^{(w)}(t_j)$ in (4) is unbiased for the data on the worker node w and has the large-sample approximate variance as follows:

$$\widehat{\mathrm{Var}}(\widehat{\mathrm{mcf}}^{(w)}(t)) = \sum_{i=1}^{r^{(w)}} \left\{ \sum_{k=1}^{j} \frac{\delta_i(t_k)}{\delta_{\cdot}^{(b)}(t_k)} \left(d_i(t_k) - \bar{d}^{(b)}(t_k) \right) \right\}^2 . \tag{6}$$

Hence, assuming that individual systems are independent conditioning on the covariates, $\widehat{\mathrm{MCF}}(t)$ in (5) is also unbiased with its variance estimated by:

$$\widehat{\mathrm{Var}}(\widehat{\mathrm{MCF}}(t)) = \frac{\sum_{m=1}^{W} \widehat{\mathrm{Var}}(\widehat{\mathrm{mcf}}^{(w)}(t))}{W^2}. \tag{7}$$

It is also possible to obtain the covariance of $\widehat{\text{MCF}}(t_j)$ and $\widehat{\text{MCF}}(t_p)$:

$$\begin{aligned}\text{Cov}(\widehat{\text{MCF}}(t_j), \widehat{\text{MCF}}(t_p)) &= \frac{1}{W^2}\text{Cov}\left(\sum_{w=1}^{W}\sum_{k=1}^{j}\bar{d}^{(w)}(t_k), \sum_{w=1}^{W}\sum_{k=1}^{p}\bar{d}^{(w)}(t_k)\right) \\ &= \text{Cov}(\widehat{\text{mcf}}^{(w)}(t_j), \widehat{\text{mcf}}^{(w)}(t_p)).\end{aligned} \tag{8}$$

Here, $\text{Cov}(\widehat{\text{mcf}}^{(w)}(t_j), \widehat{\text{mcf}}^{(w)}(t_p)) = \sum_{k=1}^{j}\sum_{k'=1}^{p}\text{Cov}(\bar{d}^{(w)}(t_k), \bar{d}^{(w)}(t_{k'}))$ and

$$\text{Cov}(\bar{d}^{(w)}(t_k), \bar{d}^{(w)}(t_{k'})) = \begin{cases} \frac{\text{Cov}(d^{(w)}(t_k), d^{(w)}(t_{k'}))}{\delta.^{(w)}(t_k)} & k < k' \\ \frac{\text{Cov}(d^{(w)}(t_k), d^{(w)}(t_{k'}))}{\delta.^{(w)}(t_{k'})} & k > k' \\ \frac{\text{Var}(d^{(w)}(t_k))}{\delta.^{(w)}(t_k)} & k = k' \end{cases} \tag{9}$$

where $\text{Var}(d^{(w)}(t_k))$ and $\text{Cov}(d^{(w)}(t_k), d^{(w)}(t_{k'}))$ are estimated by (Meeker and Escobar, 1998):

$$\widehat{\text{Var}}(d^{(w)}(t_k)) = \sum_{i=1}^{n^{(w)}}\frac{\delta_i(t_k)}{\delta.^{(w)}(t_k)}(d_i(t_k) - \bar{d}^{(w)}(t_k))^2, \quad t_k < t_{k'}, \tag{10}$$

$$\widehat{\text{Cov}}(d^{(w)}(t_k), d^{(w)}(t_{k'})) = \sum_{i=1}^{n^{(w)}}\frac{\delta_i(t_{k'})}{\delta.^{(w)}(t_{k'})}(d_i(t_{k'}) - \bar{d}^{(w)}(t_{k'}))d_i(t_k), \quad t_k < t_{k'}. \tag{11}$$

2.2.2 The Splitting Rule

The splitting rule for a node maximizes the difference between the MCF constructed on the two daughter nodes; see (3). The RF-R algorithm adopts the following splitting approaches.

- (Splitting based on the log-rank test). For any parent node which contains a number of k pooled failure times, $t_1 < t_2 < ... < t_k$, let

$$Z(t_i) = \widehat{\text{MCF}}^{(-)}(t_i) - \widehat{\text{MCF}}^{(+)}(t_i), \quad i = 1, 2, ..., k \tag{12}$$

be the difference of the MCF estimates at time t_i between the left and right daughter nodes. It follows from (7) that $\text{Var}(Z(t_i)) = \text{Var}(\widehat{\text{MCF}}^{(-)}(t_i)) + \text{Var}(\widehat{\text{MCF}}^{(+)}(t_i))$ and $\text{Cov}(Z(t_i), Z(t_j)) = \text{Cov}(\widehat{\text{MCF}}^{(-)}(t_i), \widehat{\text{MCF}}^{(-)}(t_j)) + \text{Cov}(\widehat{\text{MCF}}^{(+)}(t_i), \widehat{\text{MCF}}^{(+)}(t_j))$ for $i, j = 1, 2, ..., k$ and $i \neq j$. Let $\boldsymbol{\Sigma}$ be the covariance matrix of $\boldsymbol{Z} = (Z(t_1), Z(t_2), ..., Z(t_k))$, the test statistic is given by the quadratic form $\boldsymbol{Z}\boldsymbol{\Sigma}\boldsymbol{Z}^T$ (Klein and Moeschberger, 2005). Since the MCF estimate is asymptotically normal (Meeker and Escobar, 1998), the test statistic has a chi-squared distribution for large samples with $k-1$ degree of freedom, under the null hypothesis that $\widehat{\text{MCF}}^{(-)}(t) = \widehat{\text{MCF}}^{(+)}(t)$.

- (Splitting based on the L^2 distance). The optimum splitting essentially maximizes some distance between $\widehat{\text{MCF}}^{(-)}(t)$ and $\widehat{\text{MCF}}^{(+)}(t)$ on the two daughter nodes. Hence, a natural distance measure between two real-valued functions in the L^2 space is given by:

$$||\widehat{\text{MCF}}^{(-)}(t) - \widehat{\text{MCF}}^{(+)}(t)||_2 = \left(\int_0^{t_k} \left(\widehat{\text{MCF}}^{(-)}(t) - \widehat{\text{MCF}}^{(+)}(t) \right)^2 dt \right)^{\frac{1}{2}} \approx \left(\sum_{i=1}^{k} Z^2(t_i) \right)^{\frac{1}{2}}. \tag{13}$$

  The square root of the log-rank test statistic reduces to (13) when $\boldsymbol{\Sigma}$ becomes an identify matrix. Our numerical experiment shows that the two rules generate comparable performance in terms of accuracy. However, the splitting based on the L^2 distance is much faster as it does not require the evaluation of the covariance matrix in the log-rank test statistic.

2.2.3 Out-of-Bag (OOB) Error

The use of OOB samples is an important feature under the framework of RF. The MCF of an OOB system is constructed by only averaging those trees in which this particular system does not appear, thus the OOB prediction error is close to that of a cross-validation. The stabilization of the OOB error also helps to determine the number of trees in a forest.

Let $\gamma_i(b) = 1$ if system i is not contained in the bootstrap sample b, otherwise, $\gamma_i(b) = 0$. Then, the OOB ensemble MCF for system i can be calculated as:

$$\widehat{\text{MCF}}^{(\text{OOB})}(t; \boldsymbol{x}) = \frac{\sum_{b=1}^{B} \gamma_i(b) \sum_{j=1}^{M} \widehat{\text{MCF}}_{h_j^{(b)}}(t) I\{\boldsymbol{x} \in h_j^{(b)}\}}{\sum_{b=1}^{B} \gamma_i(b)}. \tag{14}$$

The quantification of prediction error depends on specific applications. For example, if the MCF at a particular time is of interest, the prediction error can be measured by the Mean Squared Error of the predicted MCF at that time. In this paper, we follow the idea of Ishwaran et al. (2008) and choose a more general error measure, known as the Harrell's concordance index (or, C-index). The C-index was firstly proposed in Harrell et al. (1982) for evaluating the amount of information a medical test provides about individual patients; also see Brentnall and Cuzick (2018). For the problem considered in this paper, the C-index can be interpreted as the empirical probability of correctly ranking any two systems in terms of their reliability, and can be calculated in the following way: 1) form all pairs of systems from the OOB sample, and the total number of system pairs is $\binom{n^{\text{OOB}}}{2}$ with $n^{\text{OOB}} = \sum_{b=1}^{B} \gamma_i(b)$ being the total number of OOB samples; 2) for each pair i, rank the two systems based on some reliability measures, such as the mean intensity, the cumulative number of failures up to a given time, etc. These reliability measures can be calculated from the observed failure times; 3) for each pair i, rank the two systems based on the same reliability measure, which is calculated from the predicted MCF. If the predicted rankings are consistent with the observed rankings, let $C_i = 1$ for pair i, otherwise $C_i = 0$; and 4) the C-index for an OOB sample is given by $\binom{\text{OOB}}{2}^{-1} \sum C_i$.

2.3 Theoretical Results

Some useful properties of the RF-R estimator can be obtained by extending the existing work (Fleming and Harrington, 1991; Anderson et al., 1993; Ishwaran and Kogalur, 2010). As discussed in Section 2, this paper considers the standardized covariate space $\mathbb{X}$ which is a multi-dimensional cube $[0,1]^p$. Ishwaran and Kogalur (2010) showed the uniform consistency of RSF for right-censored time-to-failure data under the assumption that the covariate space $\mathbb{X}$ has finite cardinality and the covariate $\boldsymbol{X}$ for any system is randomly sampled from a discrete covariate space $\mathbb{X}$ according to the marginal distribution μ, i.e., $\mu(A) = \mathbb{P}(\boldsymbol{X} \in A)$ for some subset A of $\mathbb{X}$. Adopting the same assumption and for any covariate j, $j = 1, 2, ..., p$, the continuous covariate space $[0,1]$ is discretized to a large number of L_j equal-width bins in order to ensure high granularity. In numerical computation, even if the covariate is continuous, discretizing covariates is always needed when splitting a tree node. The proposition below shows the uniform consistency of the ensemble RF-R estimator.

Proposition 1 (uniform consistency). *Let $v(\cdot;\boldsymbol{x})$ be the strictly positive recurrence rate given the covariate $\boldsymbol{x}$, and let $\tau = \min(\tau(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{X})$ with $\tau(\boldsymbol{x}) = \sup\{t : \int_0^t v(s;\boldsymbol{x})ds < \infty\}$. If $\mathbb{P}\{C > c\} > 0$ for $c \in [0, \tau)$, then, for $t \in (0, \tau)$ and all $\epsilon > 0$ we have*

$$\lim_{N\to\infty} \mathbb{P}\left\{ \sup_{s\in[0,t]} \left| \mathbb{E}_{\boldsymbol{X}}(\widehat{\mathrm{MCF}}^{(*)}(s;\boldsymbol{X})) - \mathbb{E}_{\boldsymbol{X}}(\mathrm{MCF}(s;\boldsymbol{X}))\right| > \epsilon \right\} = 0. \tag{15}$$

The proof is provided in the Appendix. Recall that, the fundamental idea behind RF is to reduce the variance through bagging and de-correlating trees. For a given covariate $\boldsymbol{x}$, it is well-known that the variance of the ensemble trees is given by (Hastie et al., 2009)

$$\rho(\boldsymbol{x})\mathrm{Var}(\widehat{\mathrm{MCF}}(t;\boldsymbol{x})) + B^{-1}(1-\rho(\boldsymbol{x}))\mathrm{Var}(\widehat{\mathrm{MCF}}(t;\boldsymbol{x}))$$

where $\mathrm{Var}(\widehat{\mathrm{MCF}}(t;\boldsymbol{x}))$ is the sampling variance of any single randomly drawn tree, and $\rho(\boldsymbol{x})$ is the correlation between a pair of randomly chosen trees for a given covariate $\boldsymbol{x}$. Hence, when $B \to \infty$, the variance of the ensemble trees becomes $\rho(\boldsymbol{x})\mathrm{Var}(\widehat{\mathrm{MCF}}(t;\boldsymbol{x}))$.

In Hastie et al. (2009) (page 599), the authors provided a simple approach which allows us to numerically investigate the correlation $\rho(\boldsymbol{x})$ between pairs of trees drawn from a forest, while Proposition 2 below gives the asymptotic variance of a single randomly drawn tree.

Proposition 2. *Let $\delta_{\boldsymbol{x}}(t) = 1$ if there is at least one system which is still under observation at time t with covariate $\boldsymbol{x}$, and $\delta_{\boldsymbol{x}}(t) = 0$ otherwise, and let $\widetilde{\mathrm{MCF}}(t;\boldsymbol{x}) = \int_0^t I\{\delta_{\boldsymbol{x}}(t) > 0\}d\mathrm{MCF}(t;\boldsymbol{x})$ and $\widetilde{\mathrm{MCF}}(t;\boldsymbol{X}) = \sum_{x\in\mathbb{X}} \widetilde{\mathrm{MCF}}(t;\boldsymbol{x})$. Then, for $t \in (0,\tau)$ with τ being defined in Proposition 1, the asymptotic variance of a randomly drawn tree $\widehat{\mathrm{MCF}}(t;\boldsymbol{X})$ is given by:*

$$\mathrm{Var}\left(\sqrt{n}(\widehat{\mathrm{MCF}}(t;\boldsymbol{X}) - \widetilde{\mathrm{MCF}}(t;\boldsymbol{X}))\right) = \sum_{x} \phi(\boldsymbol{x}, t) \tag{16}$$

where $\phi(\boldsymbol{x}, t) = \int_0^t \pi^{-1}(s)(1 - \Delta\mathrm{MCF}(s;\boldsymbol{x}))d\mathrm{MCF}(s;\boldsymbol{x}))$, and $\pi(\cdot)$ is the density function of the random censoring time, and $\Delta\mathrm{MCF}(t;\boldsymbol{x}) \equiv \mathrm{MCF}(s;\boldsymbol{x}) - \lim_{s\uparrow t}\mathrm{MCF}(s;\boldsymbol{x})$.

The proof is available in the Appendix.

3 Extended RF-R to Incorporate Dynamic Covariates

In this section, we show that the RF-R algorithm can be extended to handle both static system attributes and dynamic sensor measurements. For any system i, in addition to the event times $\boldsymbol{y}_i$ and a vector of static covariates $\boldsymbol{x}_i$, we also observe a q-dimensional time series, $\boldsymbol{z}_i(t) = (z_{i,1}(t), z_{i,2}(t), ..., z_{i,q}(t))$, where $z_{i,j}(t)$ can be the jth sensor measurement of some operating and environmental condition at time t.

Incorporating dynamic covariates into a tree-based method is challenging. As shown in Section 5, systems in the field can experience distinctive operating and environmental conditions, which typically have cumulative effects on the failure process. Hence, we extend the RF-R algorithm to include dynamic covariates based on the following strategy: the split of a tree node is based on the static system attributes, while the data on each node are modeled by a parametric model incorporating the dynamic covariates. In other words, we consider the scenario where the subpopulations among a large fleet of systems are mainly characterized by the static system attributes, while the dynamic sensor measurement is used for explaining the system-to-system variation of the failure processes among systems sharing similar attributes. In particular, for a vector of covariates $\boldsymbol{x} \in h_j^{(b)}$, the corresponding intensity function of the failure process is given by:

$$\widehat{\lambda}^{(b)}(t; \boldsymbol{x}) = \sum_{j=1}^{M} \widehat{\lambda}_{h_j^{(b)}}(t) I\{\boldsymbol{x} \in h_j^{(b)}\}, \tag{17}$$

where $\widehat{\lambda}_{h_j^{(b)}}(t)$ is the intensity function estimated using data from the terminal node $h_j^{(b)}$, and the ensemble intensity, $\widehat{\lambda}^{(*)}(t; \boldsymbol{x})$, is computed by averaging over the B trees:

$$\widehat{\lambda}^{(*)}(t; \boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} \widehat{\lambda}^{(b)}(t; \boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B}\sum_{j=1}^{M} \widehat{\lambda}_{h_j^{(b)}}(t) I\{\boldsymbol{x} \in h_j^{(b)}\}. \tag{18}$$

Hence, when the dynamic covariates are incorporated, the intensity functions need to be estimated on a tree node instead of estimating MCF. Let $\tilde{h}$ denote either an intermediate or terminal node, we assume that the failure process on node $\tilde{h}$ can be modeled by a NHPP with the intensity function $\lambda_{\tilde{h}}(t)$ being parameterized by a vector $(\beta_0^{\tilde{h}}, \beta_1^{\tilde{h}}, ..., \beta_q^{\tilde{h}})^T$ through a log-linear specification:

$$\log \lambda_{\tilde{h}}(t) = \beta_0^{(\tilde{h})} + \sum_{j=1}^{q} z_j(t)\beta_j^{(\tilde{h})}. \tag{19}$$

Then, $\lambda_{\tilde{h}}(t)$ can be estimated by minimizing the negative log-likelihood function with the L^1 penalty (Lasso):

$$-\log L(\beta_0^{\tilde{h}}, \boldsymbol{\beta}^{\tilde{h}}) + \omega||\boldsymbol{\beta}^{\tilde{h}}||_1 \tag{20}$$

where $\boldsymbol{\beta}^{\tilde{h}} = (\beta_1^{\tilde{h}}, \beta_2^{\tilde{h}}, ..., \beta_q^{\tilde{h}})^T$ and $\log L(\beta_0^{\tilde{h}}, \boldsymbol{\beta}^{\tilde{h}}) = \sum_{i=1}^{n^{(\tilde{h})}}(-\int \lambda_{\tilde{h}}(s)ds + \sum_{j=1}^{k} \lambda_{\tilde{h}}(t_j))$. Here, $n^{(\tilde{h})}$ and k are respectively the number of systems and pooled failure times on node $\tilde{h}$. The L^1 regularization shrinks some coefficients to be exactly zero. Although an increasingly larger number of covariates has been made available by the advances of sensing technologies, it is evident from our industry practice that many covariates are in fact redundant from either the statistical modeling or domain knowledge perspectives. Also note that, the likelihood (20) can be evaluated in parallel for distributed datasets as the contribution to the total likelihood from each system can be calculated independently.

The extended RF-R algorithm is summarized in Algorithm 3. The computational complexity of the extended RF-R algorithm depends on the dimension, q, of the dynamic covariates as well as the optimization methods used for maximizing the likelihood (20). For example, the quasi-Newton method (the variable metric algorithm) is used in our numerical examples, and the computational complexity of the extended RF-R algorithm is $\mathcal{O}(mnq^2 \log n)$, where $\mathcal{O}(q^2)$ is the cost associated with the quasi-Newton method.

The extended RF-R algorithm also satisfies the basic scalability, reliability and security requirements for big data analytics and no transfer of the raw data is required during the analysis; see Section 2. In Step 1, the data included in a bootstrap sample are generated locally on each *worker* node in parallel. In Step 2, a tree is grown until the termination condition is met. At each node and for each splitting, the *driver* node generates m covariates at random from the p covariates, and creates the candidate split point for each of the selected

m covariates. The *worker* node performs the splitting of the local bootstrap data into two subpopulations, and computes the total likelihood for both subpopulations. The computed total likelihood, using data stored on the worker node, is returned to the *driver* node. Then, the *driver* node aggregates the local likelihood constructed from all worker nodes, and the optimum splitting covariate and split-point is carried out on the *driver* node. The tree structure is updated and recorded on the *driver* node until the termination condition is met.

Algorithm 3: The extended RF-R algorithm with static and dynamic covariates

```
Data: $\boldsymbol{y}_i$, $\boldsymbol{x}_i$ and $\boldsymbol{z}_i(t)$ for all $i$
Step 1 Draw $B$ bootstrap samples from the original data.
Step 2 for $b$=1,...,$B$ do
    Grow a random tree by recursively repeating the following steps at each node of
     the tree:
    2.1) Select $m$ covariates at random from the $p$ covariates.
    2.2) Pick the best covariate and its split-point among the $m$ selected covariates
     by maximizing the $L^2$ distance of the intensity functions between the two
     daughter nodes; see (21). At each daughter node, the intensity function, which
     depends on the dynamic covariates, is estimated by minimizing the negative
     log-likelihood function with the $L^1$ penalty (20)
    2.3) Grow the tree to full size under the condition that each terminal node
     contains at least $d_0$ systems with at least one failure.
Step 3 Obtain the ensemble estimator by averaging all trees.
Step 4 Compute the prediction error using the OOB data.
```

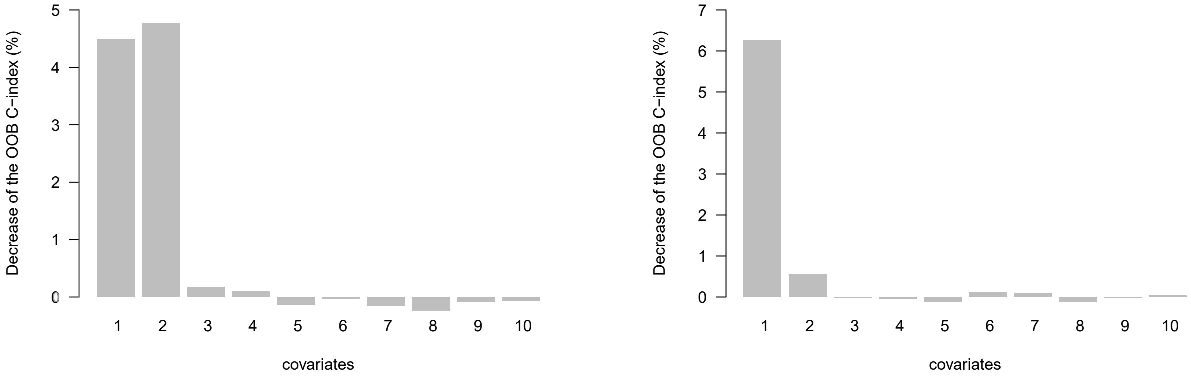
Based on the same idea described in Section 2.2.2, the splitting rule for a node maximizes distance between two real-valued intensity functions in the L^2 space:

$$||\hat{\lambda}^{(-)}(t) - \hat{\lambda}^{(+)}(t)||_2 = \left(\int_0^{t_k} \left(\hat{\lambda}^{(-)}(t) - \hat{\lambda}^{(+)}(t) \right)^2 dt \right)^{\frac{1}{2}} \tag{21}$$

where $\hat{\lambda}^{(-)}(t)$ and $\hat{\lambda}^{(+)}(t)$ respectively denote the estimated intensity functions on the two daughter nodes. The OOB ensemble intensity can also be calculated as:

$$\hat{\lambda}^{(\mathrm{OOB})}(t; \boldsymbol{x}) = \frac{\sum_{b=1}^{B} \gamma_i(b) \sum_{j=1}^{M} \hat{\lambda}_{h_j^{(b)}}(t) I\{\boldsymbol{x} \in h_j^{(b)}\}}{\sum_{b=1}^{B} \gamma_i(b)} \tag{22}$$

where $\gamma_i(b) = 1$ if system i is not contained in the bootstrap sample b, otherwise, $\gamma_i(b) = 0$. The OOB error can be measured using the C-index as described in Section 2.2.3.



(a) (b)

Figure 3: Covariate importance based on `DATASET A` and `DATASET B`

4 Numerical Examples

Section 4 investigates two illustrative examples in order to generate some critical insights and demonstrate the key advantages of the proposed RF-R. The motivating example is re-visited in Section 5 to illustrate the application of RF-R on a real industrial problem. All data sets and computer code are available on Github (`https://github.com/dnncode/RF-R`).

4.1 Numerical Example 1

We start with a simple numerical example involving 200 systems. For each system, a number of 10 system attributes are respectively sampled from the unit interval $[0, 1]$. Let $x_{i,j}$ represent the value of covariate j associated with system i ($i = 1, 2, ..., 200$, $j = 1, 2, ..., 10$), the failure data of system i are simulated from a Homogeneous Poisson Process (HPP) with the following intensity: $\lambda_i = 0.01$ if $\leq x_{i,1}, x_{i,2} \leq 0.5$, $\lambda_i = 0.1$ if $0.5 < x_{i,1}, x_{i,2} \leq 1$, otherwise $\lambda_i = 0.05$. This data set is referred to as `DATASET A`. Note that, covariates $x_3, ..., x_{10}$ are purposely made redundant in this example.

Following the classical idea of identifying variable importance under the framework of RF, Figure 3(a) shows the covariate importance as the average decrease of the OOB prediction C-index after the values of a particular covariate have been randomly permuted. A larger decrease of the OOB prediction C-index naturally indicates a stronger impact of a particular covariate on the prediction performance. It is immediately seen that, the algorithm successfully identifies covariates x_1 and x_2 as important covariates, while covariates $x_3, x_4, ..., x_{10}$ have little impact on the OOB prediction C-index and need to be excluded from the model.

We re-run the RF-R algorithm retaining only covariates x_1 and x_2. Since the space of the two covariates, x_1 and x_2, is a unit square $[0, 1]^2$, it is possible to visualize the binary partition of the covariate space by each tree. Figure 4 shows the binary partition of the covariate space $[0, 1]^2$ by eight randomly selected trees from the ensemble forests (with 500 trees). Note that, each tree divides the covariate space into a number of partitions, and each partition is represented by a terminal node. Hence, for each partition of a tree, the MCF is estimated using the data within that partition and is also shown in Figure 4.

For `DATASET A`, the intensity of the HPP falls into three classes depending on the values of x_1 and x_2. All trees in Figure 4 perform good binary partitions of the covariate space $[0, 1]^2$. For example, tree #84 partitions $[0, 1]^2$ into 13 partitions (i.e., 13 terminal nodes). The first class, $x_1, x_2 \in [0, 0.5]$, is captured by terminal nodes 1 and 2; the second class, $x_1, x_2 \in (0.5, 1]$, is reasonably represented by terminal nodes $11 \sim 13$; while the third class is captured by the remaining terminal nodes. For another example, tree #351 contains 12 terminal nodes. The first class, $x_1, x_2 \in [0, 0.5]$, is captured by terminal nodes $1 \sim 3$, and the second class, $x_1, x_2 \in (0.5, 1]$, is almost perfectly represented by terminal nodes $10 \sim 12$.

Interestingly, Figure 4 may suggest the necessity to control the depth of the tree by combining some of the terminal nodes (Huo et al., 2006). For tree #351, for example, the ideal case is to combine terminal nodes $1 \sim 3$ to form one bigger node. Note that, if the RF-R algorithm chooses the optimum splitting covariate and split point based on the log-rank test, one possible way to prune the tree is to stop further splitting a node if the minimum p-value is larger than a certain threshold, say, 0.05 or 0.1. This strategy effectively stops the tree from growing too deep. Alternatively, one might use a larger value of d_0 in the RF-R algorithm, which is the minimum number of systems with failures in a node. Fortunately,

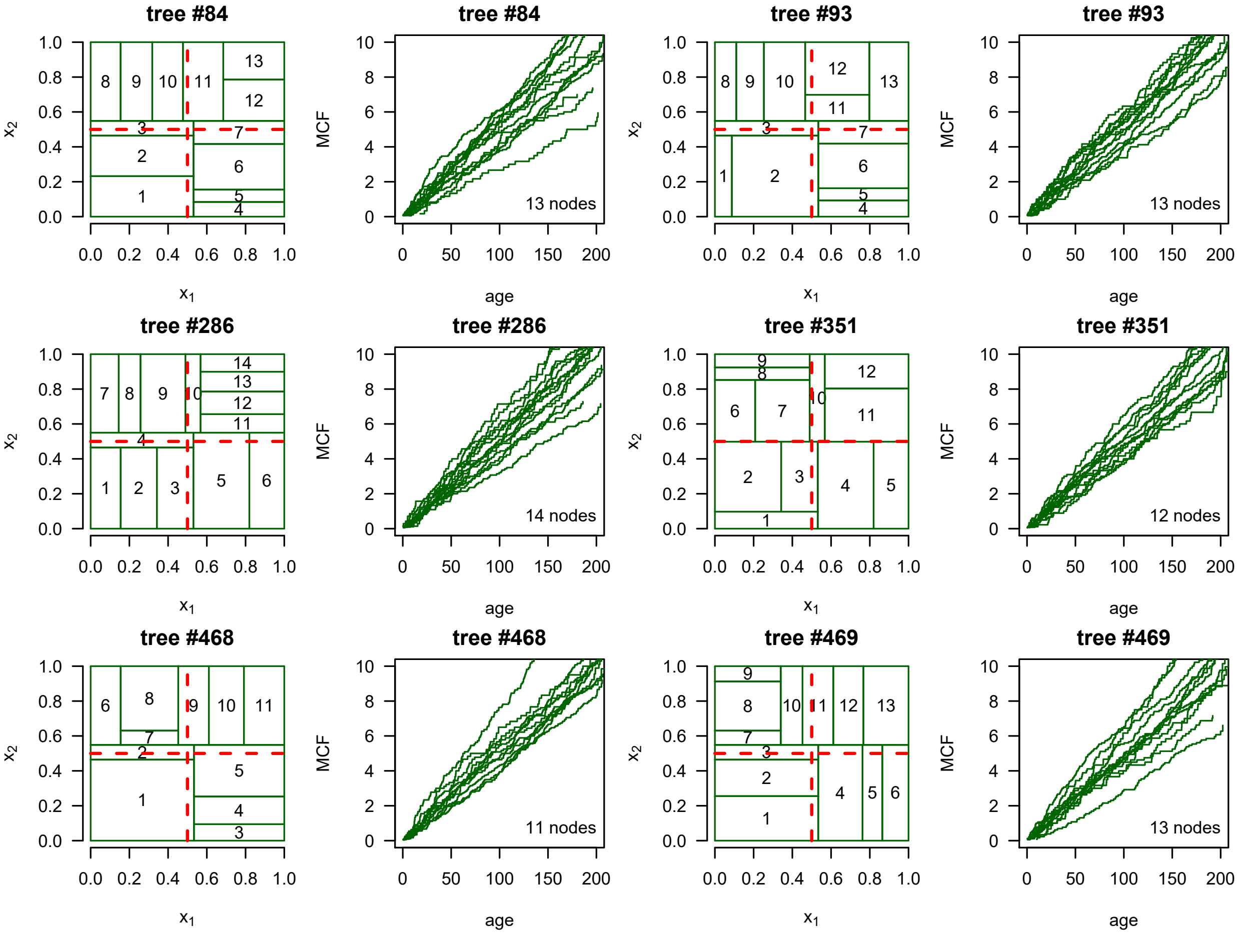


Figure 4: Columns 1 and 3 show the binary partitions of the covariate space $[0, 1]^2$ by eight randomly selected trees from the ensemble forests with the red dash lines indicating the true classification of the failure intensity. Columns 2 and 4 show the estimated MCF associated with each terminal node for the eight randomly chosen trees.

using full-grown trees seldom costs much in terms of model performance (Hastie et al., 2009). When trees are grown deep, each individual tree usually has low bias with high variance. Then, the idea of bagging (averaging full-grown trees) effectively reduces the variance of the ensemble estimator, and high model accuracy can thus be achieved.

To demonstrate the advantages of the proposed approach, a cross-validation-based comparison study is performed considering the following four candidate approaches: 1) *RF-R*: the proposed RF-R approach; 2) *MCF*: the nonparametric estimation for MCF without utilizing any system attributes; 3) *MCF-K*: the nonparametric estimation for MCF using only the data from the K nearest "neighboring" a system. Here, the distance between two systems is given by the Euclidean distance based on their covariates; and 4) *HPP*: the HPP model with a log-linear intensity, $\log(\lambda_i) = \beta_0 + \sum_{j=1}^{p} x_{i,j}\beta_j$ and $p = 10$.

Figure 5(a) shows the cross-validation-based comparison of the prediction C-index for the four approaches above. A number of 500 iterations are performed. Within each iteration, the training and testing data sets are randomly split according to a 75%-25% ratio. We see that the proposed RF-R approach generates the highest prediction C-index. Note that, C-index is an evaluation metric based on pairwise ranking: 1) firstly, we rank the reliability of two systems based on the model; 2) then, we look at the observed reliability ranking of two systems, which is subject to uncertainty and may not always reflect the true reliability ranking of the two systems. In this sense, the C-index depends not only on the methods but also on the data itself. The comparison presented in Figure 5 is fair because different methods are applied to the same data set.

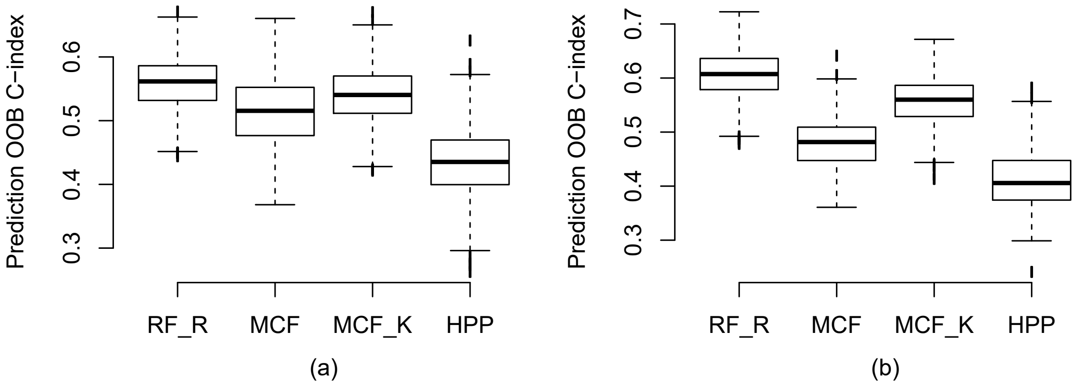


Figure 5: Cross-validation-based comparison of the prediction C-index based on `DATASET A` and `DATASET B`.

One might argue that the assumption of HPP with a log-linear intensity function is not appropriate given how `DATASET A` is simulated. Hence, we regenerate the failure data exactly from a HPP with the log-linear intensity function $\log(\lambda_i) = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2}$, where $\beta_0 = 0.01$, $\beta_1 = 2$ and $\beta_2 = 0.5$. This data set is referred to as `DATASET B`. Here, the covariates, $x_3, ..., x_{10}$, are still treated as redundant variables. The comparison results based on `DATASET B` are shown in Figure 5(b). We see that not only the RF-R method but also the two MCF-based methods outperform the parametric approach assuming HPP, even if the data are simulated exactly from a HPP. The performance of the parametric approach clearly suffers from the presence of eight redundant covariates.

Based on `DATASET B`, Figure 3(b) shows the covariate importance as the average decrease of the OOB prediction C-index. It is interesting to see that, since β_1 is set to be four times larger than β_2 when `DATASET B` is generated, the algorithm successfully identifies x_1 to be the most important covariate (all covariates are standardized). We also see that, although the importance of x_2 is much lower than that of x_1, x_2 is still obviously more important than the remaining eight redundant covariates, $x_3, ..., x_{10}$, as shown in Figure 3(b).

4.2 Numerical Example 2

We consider both static and dynamic covariates in the second numerical example. For each system, in addition to a number of 10 static covariates which are randomly sampled from the unit interval $[0, 1]$, a dynamic time-varying covariate $z(t)$ is also simulated from a Brownian motion process σB_t where B_t is a standard Brownian motion and $\sigma = 0.1$. The failure data for 200 systems are generated. For any system i, its failure times are simulated from a NHPP using the thinning method (Lewis and Shedler, 1979) with the intensity function, $\lambda_i(t) = \exp\{\beta_{i,0} + \beta_{i,1} z_i(t)\}$, where $\exp\{\beta_{i,0}\} = 0.01$ and $\beta_{i,1} = 0.5$ if $0 \leq x_{i,1}, x_{i,2} \leq 0.5$, $\exp\{\beta_{i,0}\} = \beta_{i,1} = 0.1$ if $0.5 < x_{i,1}, x_{i,2} \leq 1$, and $\exp\{\beta_{i,0}\} = 0.05$ and $\beta_{i,1} = 0$ otherwise. This data set is referred to as `DATASET C`, and the failure process depends only on the first two attributes, x_1 and x_2, as well as the time-varying covariate $z(t)$.

Figure 6 shows the importance of the ten system attributes, x_1, x_2,...,x_{10}, measured by the decrease of the OOB prediction C-index after the values of a particular attribute has been permuted. The algorithm successfully identifies the first two attributes, x_1 and x_2. Similar to Figure 4 in Example 1, Figure 7 shows how the binary partitions of the space

$[0,1]^2$ are performed by eight randomly chosen trees. For tree #3, for example, the first class, $x_1, x_2 \in [0, 0.5]$, is captured by terminal nodes 1 and 4; the second class, $x_1, x_2 \in (0.5, 1]$, is reasonably well represented by terminal nodes 10 $\sim$ 13; while the third class is captured by remaining terminal nodes. Also note that, for an ensemble approach like RF, we generally do not expect all trees performs equally good, and some "weak learners" always exist.

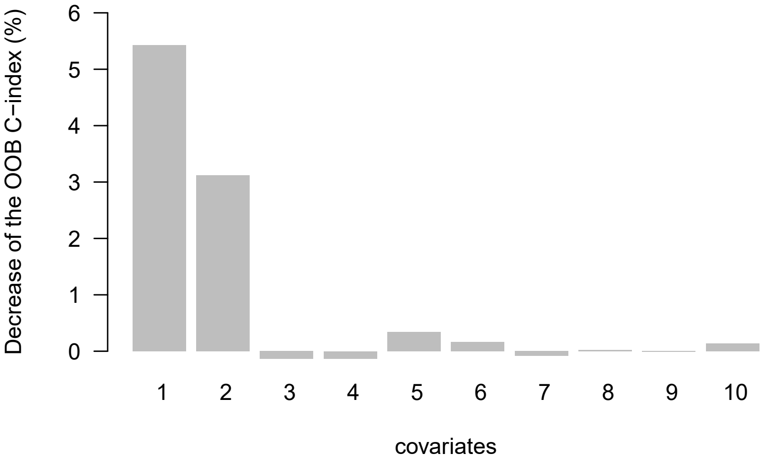


Figure 6: Covariates importance based on `DATASET C`.

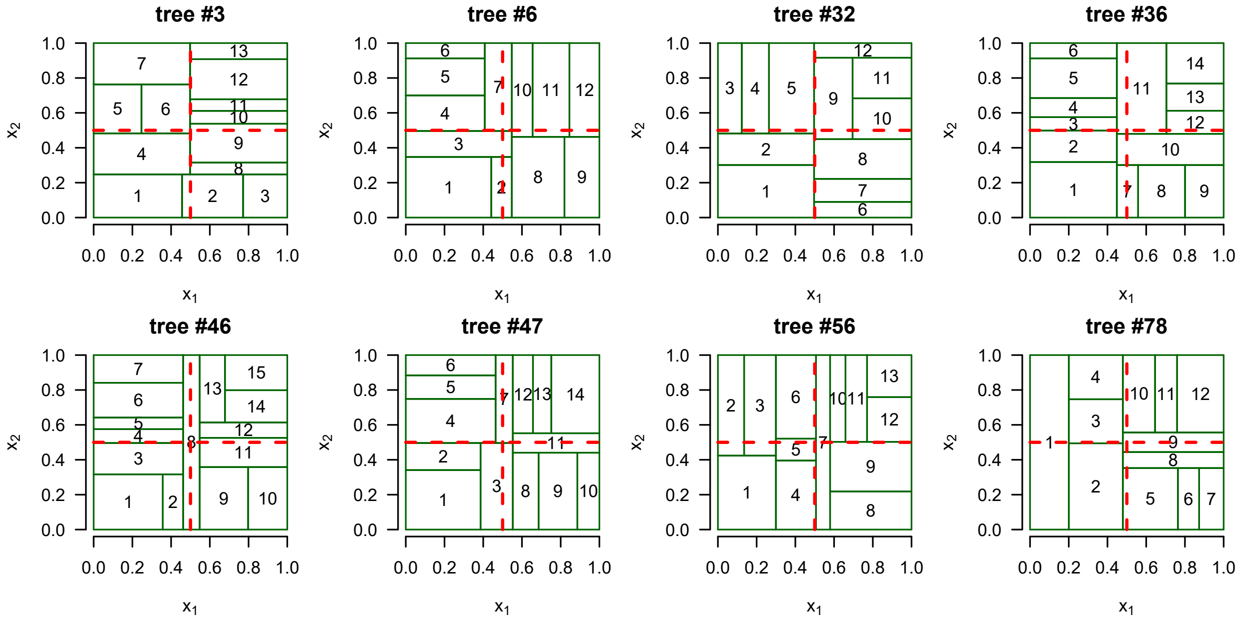


Figure 7: Binary partitions of the covariate space by eight randomly chosen trees. The red dash lines indicate the true classification of failure intensity for `DATASET C`

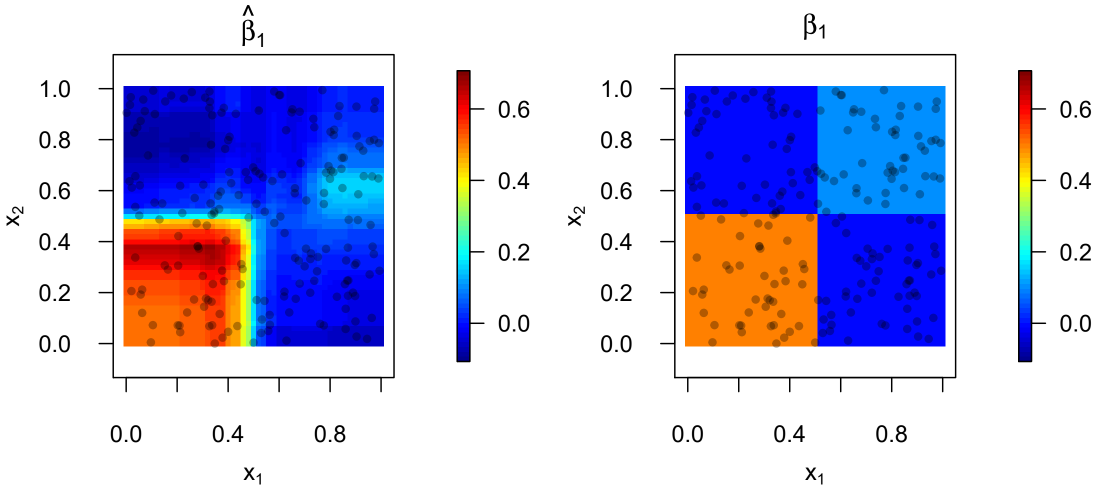


Figure 8: Left panel: an aggregated view of the estimated effect $\hat{\beta}_1$ of the dynamic covariate for different x_1 and x_2; Right panel: the true effect β_1 for `DATASET C`.

More interestingly, to show how the RF-R effectively captures the interactions between the static covariates, x_1 and x_2, and the dynamic covariate, $z(t)$, Figure 8 provides an aggregated view of the effect $\hat{\beta}_1$ from the ensemble trees over the domain $[0,1]^2$ (note that, each tree partitions the space $[0,1]^2$ into a number of rectangular areas and the effects β_1 are estimated for each area). In this figure, the panel on the left shows how the estimated effect $\hat{\beta}_1$ interacts with the static covariates, while the panel on the right shows the actual effect of β_1 (recall that the true effect β_1 of the dynamic covariate is much stronger when x_1 and x_2 are below 0.5 for `DATASET C`). We see from this figure that the proposed RF-R algorithm successfully captures the interaction between the dynamic and static covariates, which can be extremely challenging for conventional approach.

A cross-validation-based comparison of the C-index is performed for RF-R, MCF, MCF-K and NHPP which refers to the MLE assuming a NHPP model with a log-linear intensity function, $\lambda_i(t) = \exp\{\beta_{i,0} + \sum_{j=1}^{10} \beta_{i,j} x_{i,j} + \beta_{i,11} z_i(t)\}$. Figure 9(a) shows the comparison based on `DATASET C`. We see that the RF-R again outperforms in terms of the prediction C-index. Note that, since the NHPP model is not appropriate given how `DATASET C` is simulated, we re-generate the data using the intensity function, $\lambda_i(t) = \exp\{\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 z_i(t)\}$

where $\beta_0 = \log(0.01)$, $\beta_1 = 2$, and $\beta_2 = \beta_3 = 0.5$. This dataset is referred to as `Dataset D`. Based on `Dataset D`, we regenerate the comparison results as shown in Figure 9(b). The proposed RF-R still outperforms. This figure also indicates that the presence of redundant covariates $x_3 \sim x_{10}$ has a significant detrimental impact on the prediction accuracy of the NHPP model. However, the proposed RF-R algorithm appears to be much more robust against redundant covariates, which is a salient advantage of the proposed method in modern Big Data environment where the presence of redundant covariates is almost inevitable.

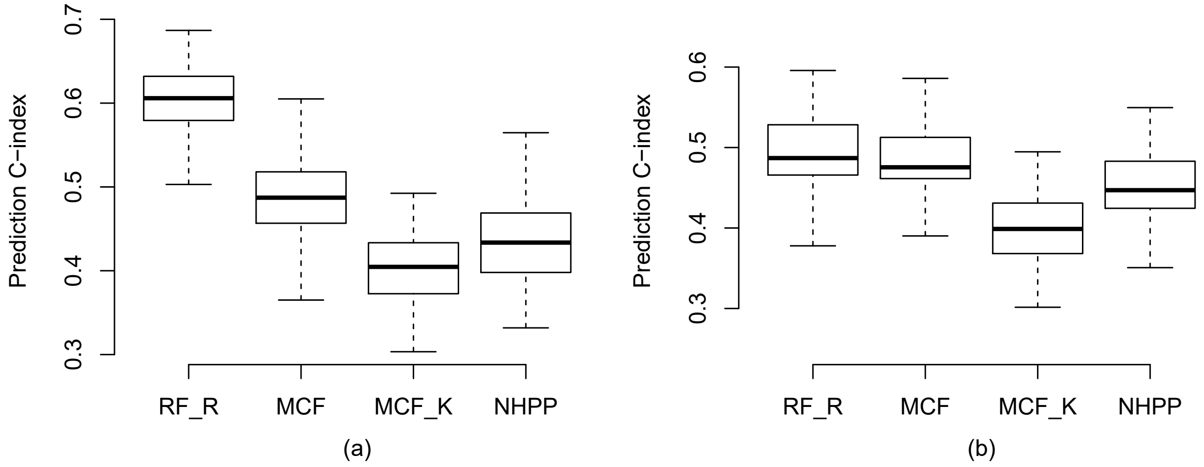


Figure 9: Cross-validation-based comparison of the prediction C-index for four different approaches. Panel (a) on the left shows the comparison results based on `Dataset C`, and panel (b) on the right shows the results based on `Dataset D`.

5 Case Study: The Motivating Example Revisited

The motivating example in Section 1.1 is re-visited in this section to demonstrate the application of the proposed approach on a real problem arising from industry. This case study is based on a modified data set which consists of 8232 oil and gas wells installed over 2007∼2017. More details can be found in Section 1.1 and the data is available from the GitHub (`https://github.com/dnncode/RF-R`).

A number of eight well attributes $x_1 \sim x_8$ are available and all covariates are standardized on the unit interval [0, 1]. In particular, the geo-locations of these wells are given by covariates x_7 (latitude) and x_8 (longitude); see Figure 1. Covariates $x_1 \sim x_6$ are static attributes that contain basic well characteristics. Figure 10 shows the standardized values of the six attributes for all 8232 systems, and the heterogeneity among systems is clearly shown.

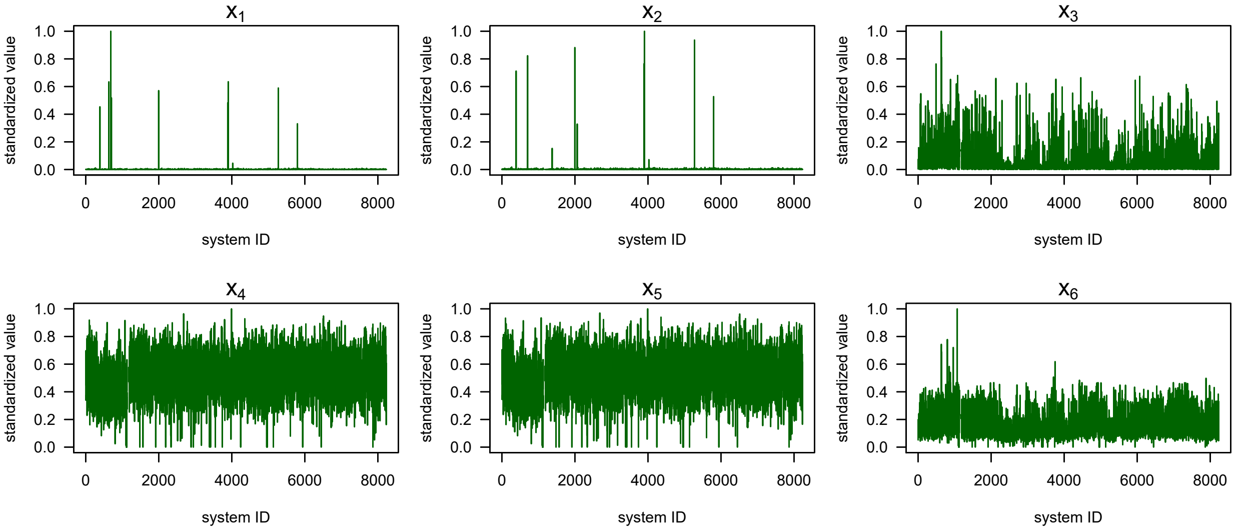


Figure 10: Standardized values of $x_1 \sim x_6$ for all 8232 systems.

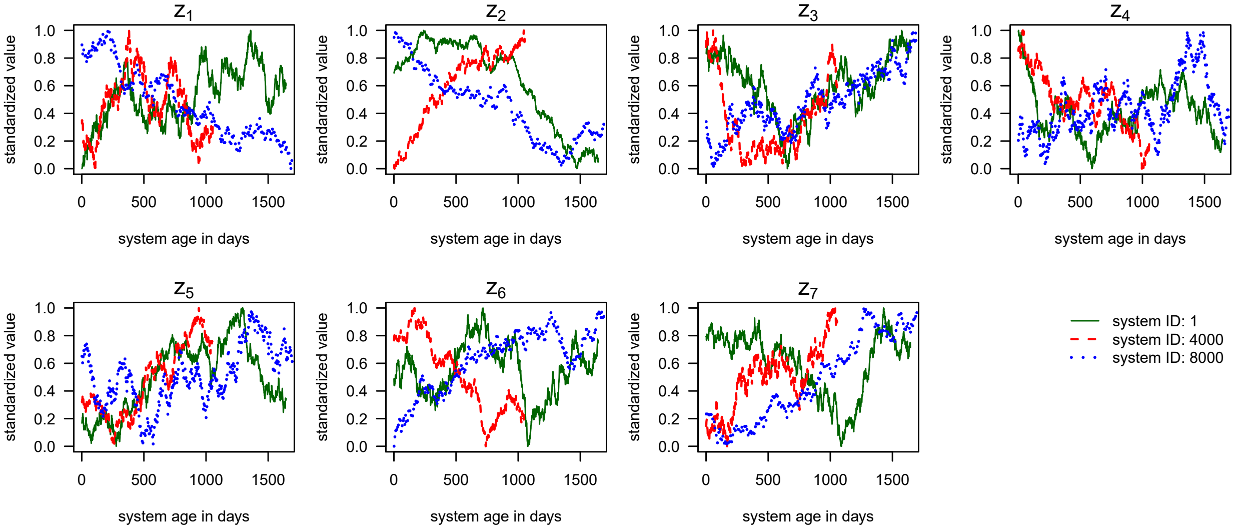


Figure 11: Sensor measurement in terms of torque, load and stress, $z_1 \sim z_7$, for three selected systems: system #1, system #4000, and system #8000.

In addition to the static covariates $x_1 \sim x_8$, seven dynamic sensor measurement of system operating conditions, $z_1(t) \sim z_7(t)$, are also available. In particular, $z_1(t) \sim z_5(t)$ are related to the dynamic torque and load measurement of each well including gear torque, load range, and so on, while $z_6(t)$ and $z_7(t)$ are respectively the monitored gearbox and structural stress. The sensor data are aggregated on daily basis. For illustrative purposes, Figure 11 plots the values of $z_1(t) \sim z_7(t)$ for three systems: system #1, #4000, and #8000. We see that, different well systems experience very different operating conditions in terms of torque, load and stress, further increasing the heterogeneity among these systems.

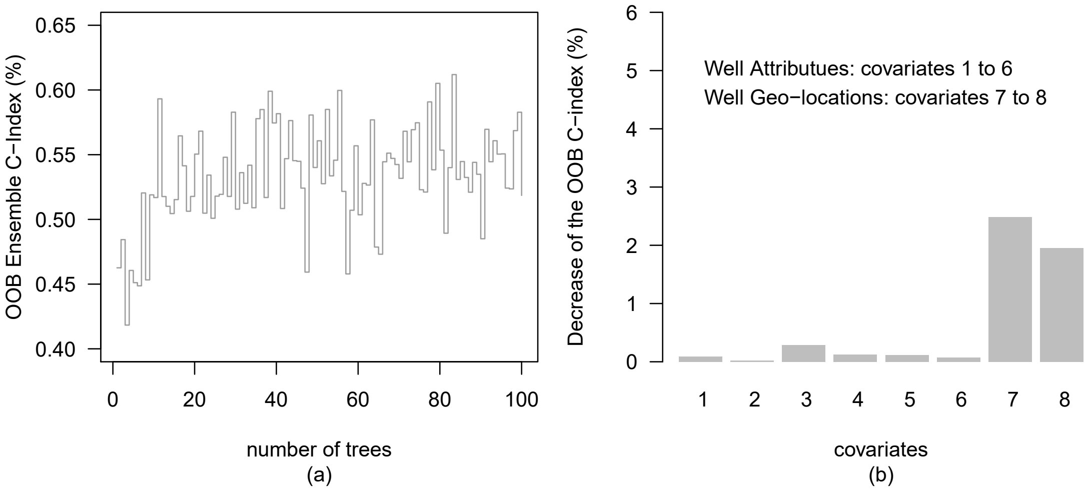


Figure 12: Panel (a) shows the OOB ensemble prediction C-index against the number of trees in a forest for the first 100 trees, and panel (b) shows the importance of the eight static system attributes measured by the average decrease of the OOB prediction C-index

We run the RF-R algorithm, and Figure 12(a) shows the OOB ensemble prediction C-index against the number of trees in a forest for the first 100 trees. It is seen that, the OOB prediction C-index quickly increases as the number of trees increases, and is stabilized approximated after 50 trees have been grown. Figure 12(b) shows the importance of the eight static system attributes as the average decrease of the OOB prediction C-index after the values of a particular covariate has been randomly permuted. Interestingly, the algorithm identifies x_7 and x_8, the geo-locations, as the two most important system attributes. Because wells at similar geographical locations share common, but unknown, environmental conditions (e.g., temperature and humidity variation, soil type, contamination, etc), geo-locations may serve as important proxies in capturing those unknown environmental factors which may lead to some important spatial patterns such as trend and clustering. The results shown in Figure 12(b) confirm that these unknown environmental factors significantly influence the system failure processes, causing distinctive failure patterns among well systems. One might also note that the importance of latitude, x_7, is slightly higher than that of longitude, x_8, indicating a stronger spatial trend along the south-north direction. This is consistent with our initial observations from Figures 1.

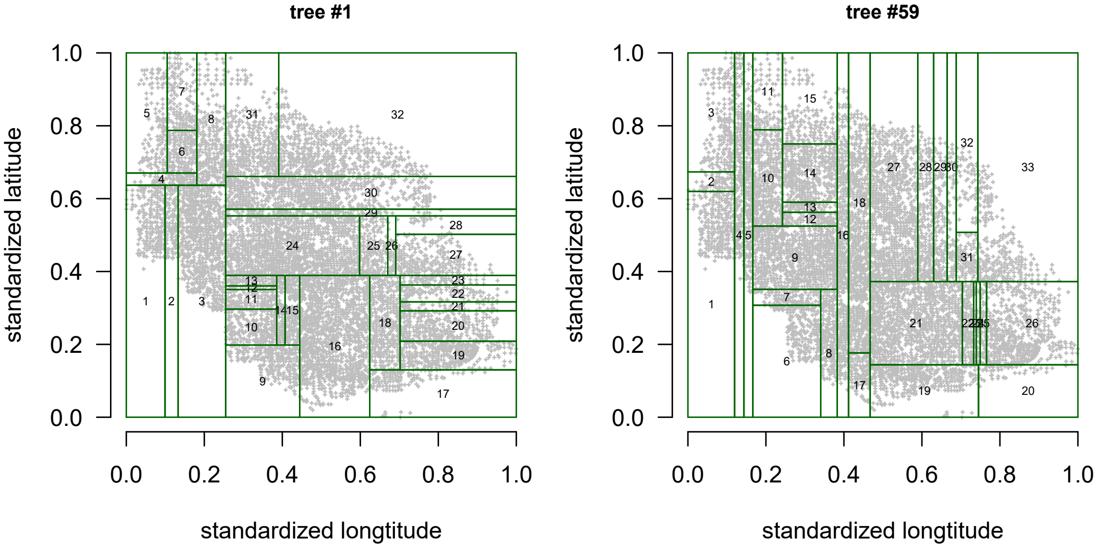


Figure 13: Binary partition of the spatial domain by two chosen trees: tree #1 and #59.

To generate more insights on how the RF-R performs, we re-run the analysis by only retaining the geo-location information, x_7 and x_8, and the sensor measurement, $z_1(t) \sim z_7(t)$. For illustrative purposes, Figure 13 shows the binary partition of the spatial domain from two chosen trees: tree #1 and #59. Tree #1 divides the spatial domain into 32 rectangular regions, while tree #59 divides the spatial domain into 33 rectangular regions. At each terminal node of a tree, the intensity function of the NHPP, which depends on the dynamic conditions measured by sensors, is estimated by minimizing the negative log-likelihood with the L^1 penalty in (20). Note that, not all sensor measurement are relevant in terms of estimating the failure process. As an illustrative example, Figure 14 shows the estimated values, $\hat{\beta}_1 \sim \hat{\beta}_7$, on every terminal node of tree #1 and tree #59. This figure is presented in the stacked view in the sense that, for each terminal node, the heights of the colored bars respectively indicate the sizes of the estimated values. It is seen that, the Lasso penalty successfully imposes sparsity on the terminal nodes of a tree: at each terminal node, only a small subset of $z_1(t) \sim z_7(t)$ is included in the estimated intensity function, which describes the failure process of the systems on that terminal node. This result also suggests that the intensity functions of systems from different geo-regions may have different dependence structures on system operational and environmental conditions, indicating a high level of heterogeneity among the 8232 systems in this case study.

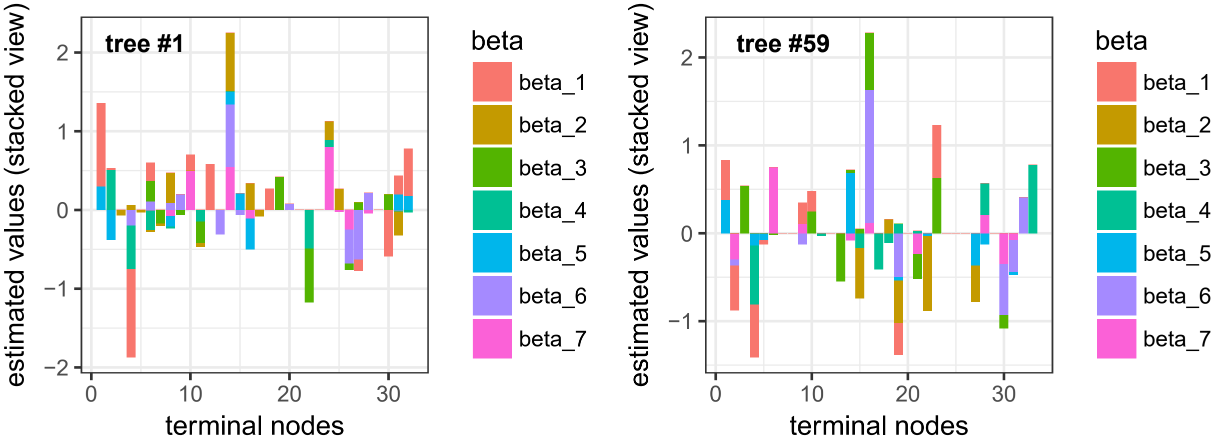


Figure 14: The estimated values, $\hat{\beta}_1 \sim \hat{\beta}_7$, on each terminal node of tree #1 and tree #59.

To further elaborate the interactions between geo-locations and system operating conditions, Figure 15 provides a spatially aggregated view of the averaged $\hat{\beta}_1 \sim \hat{\beta}_7$ from the ensemble trees over the spatial domain. Note that, each tree partitions the spatial domain $[0,1]^2$ into a number of rectangular areas and the estimates $\hat{\beta}_1 \sim \hat{\beta}_7$ are obtained for each area. It is immediately seen that the failure intensities at different geo-locations are dominated by different operational and environmental conditions. For example, z_1 appears to have a positive effect on the intensity function for systems located in the area where $x_7 > 0.7$ and $x_8 > 0.4$ (i.e., the top right area of the first subplot on the first row), z_2 generally has a negative effect on the intensity function for systems located to the area where $x_7 > 0.9$ (i.e., the top area of the second subplot on the first row), and so on. Some sensor measurements may have particularly strong local effect on the failure intensity. For example, z_4 has a strong effect on the intensity function for those systems in the area where x_7 and x_8 are respectively close to 0.4 and 0.6 (i.e., the central area of the first subplot on the second row), and z_6 has a particularly strong effect on the intensity function for the systems in the area where x_7 and x_8 are respectively close to 0.2 and 0.4, and so on. These observations yield critical insights on system reliability and can be potentially useful for system maintenance and operation.

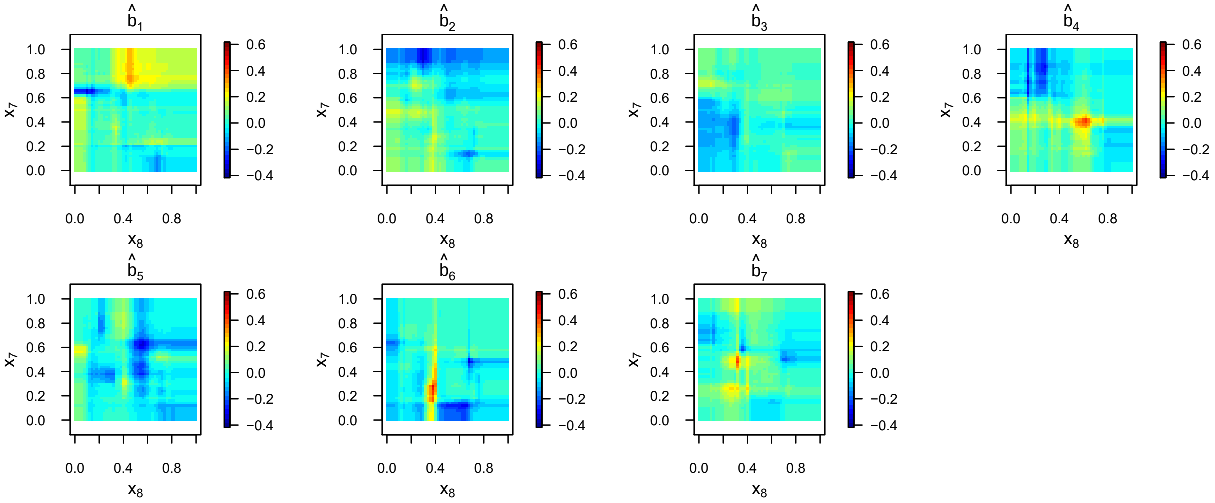


Figure 15: A spatially aggregated view of the averaged $\hat{\beta}_1 \sim \hat{\beta}_7$ over the spatial domain.

We compare the C-index, using cross-validation, between the RF-R, MCF, MCF-K, NHPP with a log-linear intensity function, and the Gamma frailty model. The Gamma frailty model is a commonly used approach to capture the random variation among systems; see Lindqvist (2006). Figure 16 shows the comparison result. The proposed RF-R again outperforms in terms of the prediction C-index. The parametric approaches, such as the NHPP and Gamma frailty model, suffer from the presence of a large number of redundant covariates. The fact that MCF has the second best performance suggests that we would rather not to use any covariate information than incorporating a large number of redundant covariates.

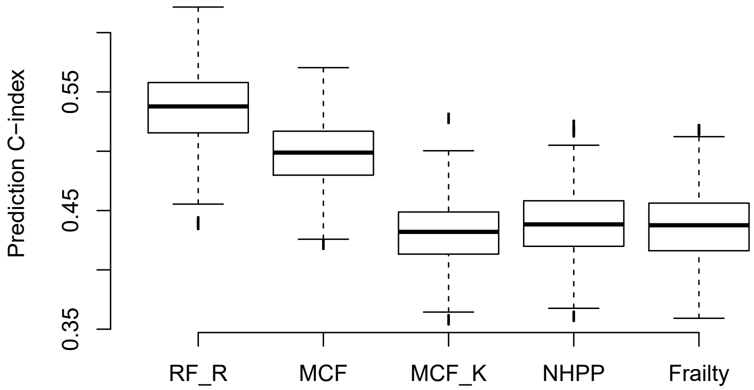


Figure 16: Cross-validation-based comparison of the prediction C-index for four different approaches for the case study

In repairable system reliability analysis, the cumulative hazard function is often of interest (Meeker and Escobar, 1998). Hence, a number of 6 systems are selected from the 8232 wells. Figure 17 shows both the true cumulative failures over time and the estimated cumulative hazard functions for the selected systems. To capture the uncertainty, this figure also includes the estimated cumulative hazard from individual trees. We see that, the RF-R effectively models the system reliability based on the recurrence data from a large fleet of repairable systems with both static and dynamic covariates.

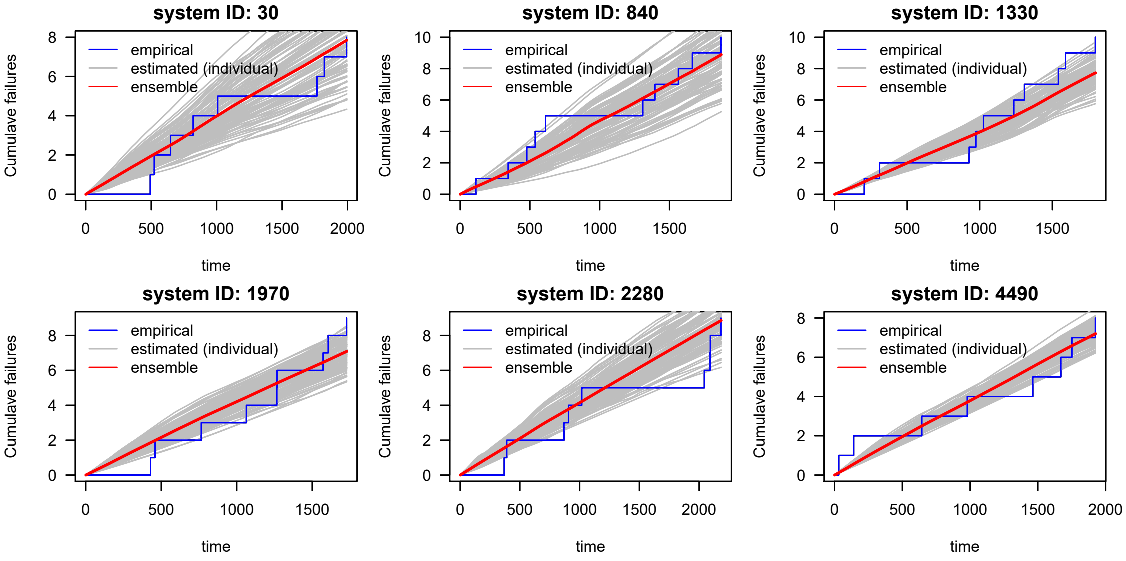


Figure 17: Estimated cumulative hazard for selected systems.

6 Conclusions

An algorithm called RF-R has been proposed for analyzing large heterogeneous repairable system reliability data with static system attributes and dynamic sensor measurement. Comprehensive numerical studies and comparison have been performed and the advantages of the proposed method have been demonstrated. The strengths of the proposed algorithm lie in the integration of the powerful Random Forests algorithm and the classical statistical reliability data analysis methodologies. This work timely addressed some pressing challenges facing reliability engineers today, including system heterogeneity, covariate selection, model specification and data locality in the modern Big Data environment. Computer program and data sets are made available on GitHub to facilitate the adoption and application of the proposed method in practice.

References

Anderson, P. K. and Borgan, O. and Gill, R. D., and Keiding, N. (1993). *Statistical Models based on Counting Processes*, Springer-Verlag, New York.

Bacchetti, P. and Segal, M. (1995). Survival Trees with Time-Dependent Covariates: Application to Estimating Changes in the Incubation Period of AIDS. *Lifetime Data Analysis*, **1**, 35-47.

Brentnall, A. and Cuzick, J. (2018). Use of the Concordance Index for Predictors of Censored Survival Data. *Statistical Methods in Medical Research*, **27**, 2359-2373.

Bou-Hamad, I. and Larocque, D. and Ben-Ameur, H. and Masse, L. and Vitaro, F. and Tremblay, R. (2009). Discrete-Time Survial Trees. *Canadian Journal of Statistics*, **37**, 17-32.

Bou-Hamad, I. (2011). A Review of Survival Trees. *Statistics Surveys*, **5**, 44-71.

Breiman, L. (2001). Random Forests. *Machine Learning*, **45**, 5-32.

Chipman, H. A. and George, E. I. and McCulloch, R. E. (2006). Bart: Bayesian Additive Regression Trees. *The Annals of Applied Statistics*, **4**, 266–298.

Doganaksoy, N. and Hahn, G. J. and Meeker, W. Q. (2006). How to Analyze Reliability Data for Repairable Products. *Quality Progress*, **39**, 93–95.

Energy Information Administration (2017). U.S. Oil and Natural Gas Wells by Production Rate. https://www.eia.gov/petroleum/wells/.

Fan, J. and Su, X. G., and Levine, R. and Nunn, M. and Leblanc, M. (2006). Trees for Censored Survival Data by Goodness of Split, with Application to Tooth Prognosis. *Journal of the American Statistical Association*, **101**, 959–967.

Fan, J. and Nunn, M. and Su, X. G. (2009). Multivariate Exponential Survial Trees and Their Application to Tooth Prognosis. *Computational Statistics and Data Analysis*, **53**,1110–1121.

Fleming, T. R. and Harrington, D. P. (1991). *Counting Processes and Survival Analysis*, John Wiley & Sons, New York.

Harrell, F. and Califf, R. and Pryor, D. and Lee, K. and Rosati, R. (2018). Evaluating the Yield of Medical Tests. *Journal of American Medicine Association*, **247**, 2543–2546.

Hastie, T. and Tibshirani, R. and Fredman, J. (2009). *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*, Springer.

Hong, Y. and Zhang, M. and Meeker, W. Q. (2018). Big Data and Reliability Applications: The Complexity Dimension. *Journal of Quality Technology*, **50**, 00–00.

Hothorn, T. and Lausen, B. and Bener, A. and Radespiel-Troger, M. (2004). Bagging survival trees. *Statistics in Medicin*, **23**, 77–91.

Hothorn, T. and Bhlmann, P. and Dudoit, S. and Molinaro, A. and Van Der Laan, M. (2006). Survival Ensembles. *Biostatistics*, **7**, 355–373.

Huo, X. M. and Kim, S. B., and Tsui, K. L. and Wang S. C. (2006). FBP: A Frontier-Based Tree-Pruning Algorithm. *INFORMS Journal on Computing*, **18**, 494–505.

Ishwaran, H. and Kogalur, U. B. and Blackstone, E. H. and Lauer, M. S. (2008). Random Survival Forests. *The Annals of Applied Statistics*, **2**, 841–860.

Ishwaran, H. and Kogalur, U. B. (2010). Consistency of Random Survival Forests. *Statistics Probability Letter*, **80**, 1056–1064.

Ishwaran, H. and Kogalur, U. B. and Gorodeski, E. Z. and Minn, A. J. and Lauer, M. S. (2010). High-Dimensional Variable Selection for Survival Data. *Journal of the American Statistical Association*, **105**, 205–217.

Klein, J. P. and Moeschberger, M. L. (2005). *Survival Analysis – Techniques for Censored and Truncated Data*, Springer.

Lewis, P. and Shedler, G. (1979). Simulation of Nonhomogenous Poisson Processes by Thinning. *Naval Research Logistics Quarterly*, **26**, 403–413.

Lindqvist, B. H. and Elvebakk, G. and Heggland, K. (2003). The Trend-Renewal Process for Statistical Analysis of Repairable Systems. *Technometrics*, **45**, 31–44.

Lindqvist, B. H. (2006). On the Statistical Modeling and Analysis of Repairable Systems. *Statistical Scienc*, **21**, 532–551.

Meeker, W. Q. and Escobar, L. A. (1998). *Statistical Methods for Reliability Data*, John Wiley & Sons, New York.

Meeker, W. Q. and Hong, Y. (2014). Reliability Meets Big Data: Opportunities and Challenges. *Quality Engineering*, **26**, 102–116.

Mittman, E. T. and Lewis-Beck, C. and Meeker, W. (2018). A Hierarchical Model for Heterogenous Reliability Field Data. *The Annals of Applied Statistics*, to appear.

Nelson, W. (1995). Confidence Limits for Recurrence Data: Applied to Cost or Number of Product Repairs. *Technometrics*, **37**, 147–157.

Pan, R. and Rigdon, S. E. (2009). Bayes Inference for General Repairable Systems. *Journal of Quality Technology*, **41**, 82–94.

Pratola, M. T. and Higdon, D. (2016). Bayesian Additive Regression Tree Calibration of Complex High-Dimensional Computer Models. *Technometrics*, **58**, 166–179.

Pulcini, G. (2001). A Bounded Intensity Process for the Reliability of Repairable Equipment. *Journal of Quality Technology*, **33**, 480–492.

Rigdon, S. E. and Basu, A. P. (2000). *Statistical Methods for the Reliability of Repairable Systems*, John Wiley & Sons, New York.

Apache Spark (2018). *Apache Spark Documentation*, https://spark.apache.org/.

Stocker, R. and Pena, E. A. (2007). *A General Class of Parametric Models for Recurrent Event Data*, **49**, 210–220.

Terenin, A. and Dong, S. F. and Draper, D. (2017). GPU-Accelerated Gibbs Sampling: A Case Study of the Horseshoe Probit Model. *arXiv*, **1608.04329v4**.

Xu, Z. and Hong, Y. and Meeker, W. and Osborn, B. E. and Illouz, K. (2017). A Multi-Level Trend-Renewal Process for Modeling Systems With Recurrence Data. *Technometrics*, **59**, 225–236.

Ye, Z. and Hong, Y. and Xie, Y. (2013). How Do Heterogeneities in Operational Environments Affect Field Failures. *The Annals of Applied Statistics*, **7**, 2249–2271.

Ye, Z. and Murthy, D. N. P. and Xie, M. and Tang, L. C. (2013). Optimal Burn-in for Repairable Products Sold with a Two-Dimensional Warranty. *IEEE Transactions on Reliability*, **45**, 164–176.

Yang, Q. and Hong, Y. and Chen, Y. and Shi, J. (2012). Failure Profile Analysis of Complex Repairable Systems With Multiple Failure Modes. *IEEE Transactions on Reliability*, **66**, 180–191.

Zuo, J. and Wu, H. and Meeker, W. Q. (2012). Asymptotic Properties of Some Estimators of the Mean Cumulative Function. *Journal of Statistical Planning and Inference*, **142**, 2943–2952.

Appendix: Proof of Propositions 1 and 2

(this section can be moved to Supplementary Materials if necessary)

Proposition 1 shows the uniform consistency of the ensemble RF-R estimator. To prove this proposition, note that,

$$\begin{aligned}
&\lim_{N\to\infty} \mathbb{P}\left\{\sup_{s\in[0,t]}\left|\mathbb{E}_{\boldsymbol{X}}(\widehat{\text{MCF}}^{(*)}(s;\boldsymbol{X})) - \mathbb{E}_{\boldsymbol{X}}(\text{MCF}(s;\boldsymbol{X}))\right| > \epsilon\right\} \\
&\quad = \lim_{N\to\infty} \mathbb{P}\left\{\sup_{s\in[0,t]}\left|\frac{1}{B}\sum_{b=1}^{B}\mathbb{E}_{\boldsymbol{X}}(\widehat{\text{MCF}}^{(b)}(s;\boldsymbol{X})) - \mathbb{E}_{\boldsymbol{X}}(\text{MCF}(s;\boldsymbol{X}))\right| > \epsilon\right\} \qquad (23) \\
&\quad \leq \lim_{N\to\infty} \mathbb{P}\left\{\frac{1}{B}\sum_{b=1}^{B}\sup_{s\in[0,t]}\left|\mathbb{E}_{\boldsymbol{X}}(\widehat{\text{MCF}}^{(b)}(s;\boldsymbol{X})) - \mathbb{E}_{\boldsymbol{X}}(\text{MCF}(s;\boldsymbol{X}))\right| > \epsilon\right\}.
\end{aligned}$$

Hence, it is sufficient to show that, for any bootstrap sample b, we have

$$\lim_{N^{(b)}\to\infty} \mathbb{P}\left\{\sup_{s\in[0,t]}\left|\mathbb{E}_{\boldsymbol{X}}(\widehat{\text{MCF}}^{(b)}(s;\boldsymbol{X})) - \mathbb{E}_{\boldsymbol{X}}(\text{MCF}(s;\boldsymbol{X}))\right| > \epsilon\right\} = 0 \qquad (24)$$

where $N^{(b)}$ is the sample size of the bootstrap sample b.

Let η_i is an indicator if sample i experiences at least one failure before the censoring time c_i, i.e., $\eta_i = 1$ if $d_i(c_i) > 0$, and $\eta_i = 0$ otherwise. Then, the law of large numbers lead to the following result:

$$\frac{1}{N^{(b)}}\sum_{i=1}^{N^{(b)}} I(\boldsymbol{X}_i \in A, \eta_i = 1) \overset{\text{a.s.}}{\to} \mathbb{P}\{\boldsymbol{X}\in A, \eta = 1\} = \mathbb{P}\{\boldsymbol{X}\in A\}\mathbb{P}\{\eta = 1\ |\boldsymbol{X}\in A\}. \qquad (25)$$

Note that, $\boldsymbol{X}$ and η are not statistically independent. However, it is reasonable to assume that $\mathbb{P}\{\eta = 1\ |\boldsymbol{X}\in A\} > 0$ for any $\boldsymbol{X}\in A$ (in other words, there exists no subset A of $\mathbb{X}$

such that the probability of observing failures is zero), (25) implies

$$I\left(\sum_{i=1}^{N^{(b)}} I(\boldsymbol{X}_i \in A, \eta_i = 1) \geq d_0\right) \overset{\text{a.s.}}{\to} 1. \tag{26}$$

Equation (26) implies that the termination rule (i.e., a terminal node must have d_0 systems with at least one failure) almost surely holds for any terminal node A. In other words, the tree almost surely has terminal nodes for all possible discretized values of $\mathbb{X}$:

$$\widehat{\text{MCF}}^{(b)}(s; \boldsymbol{X}) = \sum_{x \in \mathbb{X}} I(\boldsymbol{X} = \boldsymbol{x})\widehat{\text{MCF}}_{h^{(b)}}(s) + o_p(1). \tag{27}$$

It follows from (27) that

$$\begin{aligned}
&\sup\nolimits_{s\in[0,t]} \left|\mathbb{E}_{\boldsymbol{X}}(\widehat{\text{MCF}}^{(b)}(s; \boldsymbol{X})) - \mathbb{E}_{\boldsymbol{X}}(\text{MCF}(s; \boldsymbol{X}))\right| \\
&\quad = \sup\nolimits_{s\in[0,t]} \left|\mathbb{E}_{\boldsymbol{X}}\left(\sum_{x\in\mathbb{X}} I(\boldsymbol{X} = \boldsymbol{x})\widehat{\text{MCF}}_{h^{(b)}}(s)\right) - \mathbb{E}_{\boldsymbol{X}}(\text{MCF}(s; \boldsymbol{X})) + o_p(1)\right| \qquad (28)\\
&\quad = \sum_{x\in\mathbb{X}} \mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) \sup\nolimits_{s\in[0,t]} \left|\widehat{\text{MCF}}_{h^{(b)}}(s) - \text{MCF}(s; \boldsymbol{x})\right| + o_p(1).
\end{aligned}$$

Hence, to show the uniform convergence of $\mathbb{E}_{\boldsymbol{X}}(\widehat{\text{MCF}}^{(b)}(s; \boldsymbol{X}))$, it is sufficient to show the uniform convergence of $\widehat{\text{MCF}}_{h^{(b)}}(s)$ to $\text{MCF}(s; \boldsymbol{x})$ at each terminal node $h^{(b)}$, which requires the theorem of Anderson et al. (1993). Let $\delta_{\cdot}^{(b)}(s; \boldsymbol{x})$ be the number of systems with covariate $\boldsymbol{x}$ which are still being observed at time s, then, the Anderson's theorem says that, if the following two conditions are satisfied:

$$\int_0^t \frac{I[\delta_{\cdot}^{(b)}(s; \boldsymbol{x}) > 0]}{\delta_{\cdot}^{(b)}(s; \boldsymbol{x})} v(s; \boldsymbol{x}) ds \overset{\text{P}}{\to} 0 \tag{29}$$

and

$$\int_0^t \left(1 - I[\delta_\cdot^{(b)}(s;\boldsymbol{x}) > 0]\right) v(s;\boldsymbol{x})ds \xrightarrow{\text{P}} 0, \tag{30}$$

then,

$$\sup_{s\in[0,t]} \left|\widehat{\text{MCF}}_{h^{(b)}}(s) - \text{MCF}(s;\boldsymbol{x})\right| \xrightarrow{\text{P}} 0. \tag{31}$$

Hence, it is necessary to show that: 1) conditions (29) and (30) still hold for our problem, and 2) (29) and (30) still lead to the result shown in (31).

Firstly, to show that the two conditions (29) and (30) hold, let $n^{(b)}(\boldsymbol{x})$ be the sample size on the terminal node that contains $\boldsymbol{x}$. Then, for $s \in [0,t]$ we have,

$$\begin{aligned} \frac{1}{n^{(b)}(\boldsymbol{x})}\delta_\cdot^{(b)}(s;\boldsymbol{x}) \geq \frac{1}{n^{(b)}(\boldsymbol{x})} \sum_{i=1}^{n^{(b)}(\boldsymbol{x})} I[C_i \geq t, \boldsymbol{X}_i = \boldsymbol{x}] \\ \xrightarrow{\text{a.s.}} \mathbb{P}(\boldsymbol{X} = \boldsymbol{x})\mathbb{P}(C \geq t) > 0 \end{aligned} \tag{32}$$

where the second line on the right hand side is based on the assumptions that $\mathbb{P}\{C > c\} > 0$ for any $c \in [0,\tau)$, and $\boldsymbol{X}$ and C are independent. The inequality above implies that

$$\inf_{s\in[0,t]}\delta_\cdot^{(b)}(s;\boldsymbol{x}) \xrightarrow{\text{P}} \infty. \tag{33}$$

In other words, the number of systems with covariate $\boldsymbol{x}$ which are still being observed at time s approaches infinity when the total sample size goes to infinity, which is intuitively true. Since $v(\cdot;\boldsymbol{x})$ is bounded over $[0,t]$, it is noted that the result above is the sufficient condition for the two conditions (29) and (30) to be satisfied.

Secondly, to show that the conditions (29) and (30) still lead to the result shown in (31), we define:

$$\begin{aligned}\widehat{\mathrm{MCF}}_{h^{(b)}}(s) &= \int_0^s \delta_{\cdot}^{(b)}(u;\boldsymbol{x})d\Lambda_{h^{(b)}}(u)\\ \mathrm{MCF}_{h^{(b)}}^{(**)}(s) &= \int_0^s v(u;\boldsymbol{x})I[\delta_{\cdot}^{(b)}(u;\boldsymbol{x})>0]du \\ \mathrm{MCF}(s;\boldsymbol{x}) &= \int_0^s v(u;\boldsymbol{x})du\end{aligned} \tag{34}$$

where $\Lambda_{h^{(b)}}(u)$ denotes the number of failures in the time interval $(0,u]$ on the tree node h for the bootstrap sample b. Then, invoking the Lenglart's inequality (see (2.5.18) of Anderson et al. (1993)), for any $\eta_1,\eta_2>0$, we have

$$\begin{aligned}&\mathbb{P}\left(\sup_{s\in[0,t]}|\widehat{\mathrm{MCF}}_{h^{(b)}}(s)-\mathrm{MCF}_{h^{(b)}}^{(**)}(s)|>\eta_1\right)\\ &\le\frac{\eta_2}{\eta_1^2}+\mathbb{P}\left(\langle\widehat{\mathrm{MCF}}_{h^{(b)}}(s)-\mathrm{MCF}_{h^{(b)}}^{(**)}(s)\rangle(t)>\eta_2\right)\end{aligned} \tag{35}$$

where $\widehat{\mathrm{MCF}}_{h^{(b)}}(s)-\mathrm{MCF}_{h^{(b)}}^{(**)}(s)$ is a local square integrable martingale, and it follows from condition (29) and (4.1.5) of Anderson et al. (1993) that

$$\langle\widehat{\mathrm{MCF}}_{h^{(b)}}(s)-\mathrm{MCF}_{h^{(b)}}^{(**)}(s)\rangle(t)=\int_0^t\frac{I[\delta_{\cdot}^{(b)}(s;\boldsymbol{x})>0]}{\delta_{\cdot}^{(b)}(s;\boldsymbol{x})}v(s;\boldsymbol{x})ds\xrightarrow{\mathrm{P}}0. \tag{36}$$

Hence, the Lenglart's inequality (35) implies that

$$\sup_{s\in[0,t]}|\widehat{\mathrm{MCF}}_{h^{(b)}}(s)-\mathrm{MCF}_{h^{(b)}}^{(**)}(s)|\xrightarrow{\mathrm{P}}0.$$

Since $|\mathrm{MCF}_{h^{(b)}}^{(**)}(s)-\mathrm{MCF}(s;\boldsymbol{x})|=\int_0^s(1-I[\delta_{\cdot}^{(b)}(u;\boldsymbol{x})>0])v(u;\boldsymbol{x})du\xrightarrow{\mathrm{P}}0$ due to condition (30), we have

$$\sup_{s\in[0,t]}|\widehat{\mathrm{MCF}}_{h^{(b)}}(s)-\mathrm{MCF}(s;\boldsymbol{x})|\xrightarrow{\mathrm{P}}0.$$

as was to be proved.

Proposition 2 gives the asymptotic variance of a single randomly drawn tree. The proof requires equation (27) and Theorem 3.2.1 in Fleming and Harrington (1991). It follows from (27) that

$$
\begin{aligned}
&\mathrm{Var}(\sqrt{n}(\widehat{\mathrm{MCF}}(t;\boldsymbol{X})-\widetilde{\mathrm{MCF}}(t;\boldsymbol{X})))\\
&\qquad = \mathrm{Var}(\sqrt{n}\sum_{\boldsymbol{x}\in\mathbb{X}} I(\boldsymbol{X}=x)(\widehat{\mathrm{MCF}}_{h(\boldsymbol{x})}(t)-\widetilde{\mathrm{MCF}}(t;\boldsymbol{x}))+o_p(1))\\
&\qquad = \sum_{\boldsymbol{x}\in\mathbb{X}} \mathbb{E}(\sqrt{n}I(\boldsymbol{X}=x)(\widehat{\mathrm{MCF}}(\boldsymbol{x})(t)-\widetilde{\mathrm{MCF}}(t;\boldsymbol{x})))^2\\
&\qquad\quad - n\sum_{\boldsymbol{x}\in\mathbb{X}} I(\boldsymbol{X}=x)\mathbb{E}^2(\widehat{\mathrm{MCF}}(\boldsymbol{x})(t)-\widetilde{\mathrm{MCF}}(t;\boldsymbol{x}))+o_p(1)
\end{aligned}
\tag{37}
$$

where $h(\boldsymbol{x})$ represents a terminal node that contains $\boldsymbol{x}$.

Since $\widehat{\mathrm{MCF}}_{h(\boldsymbol{x})}(t)$ is asymptotically unbiased (on that particular node), the second and third term on the right-hand-side of (37) vanishes and we have

$$
\mathrm{Var}(\sqrt{n}(\widehat{\mathrm{MCF}}(t;\boldsymbol{X})-\widetilde{\mathrm{MCF}}(t;\boldsymbol{X}))) = \sum_{\boldsymbol{x}\in\mathbb{X}} \mathbb{E}(\sqrt{\bar{n}_{\boldsymbol{x}}}(\widehat{\mathrm{MCF}}_{h(\boldsymbol{x})}(t)-\widetilde{\mathrm{MCF}}(t;\boldsymbol{x})))^2 \tag{38}
$$

where $\bar{n}_{\boldsymbol{x}}$ is the expected number of systems with covariates $\boldsymbol{x}$. Then, it follows from Theorem 3.2.1 of Fleming and Harrington (1991) that

$$
\begin{aligned}
\mathrm{Var}\left(\sqrt{n}(\widehat{\mathrm{MCF}}(t;\boldsymbol{X})-\widetilde{\mathrm{MCF}}(t;\boldsymbol{X}))\right) &= \int_0^t \pi^{-1}(s)(1-\Delta\mathrm{MCF}(s;\boldsymbol{x}))d\mathrm{MCF}(s;\boldsymbol{x}))\\
&= \sum_{x}\phi(\boldsymbol{x},t)
\end{aligned}
\tag{39}
$$

as was to be proved.