Reliability Analysis of Artificial Intelligence Systems Using Recurrent Events Data from Autonomous Vehicles

Jie Min[1], Yili Hong[1], Caleb B. King[2], and William Q. Meeker[3]

[1]Department of Statistics, Virginia Tech, Blacksburg, VA 24061

[2]JMP Division, SAS, Cary, NC 27513

[3]Department of Statistics, Iowa State University, Ames, IA 50011

Abstract

Artificial intelligence (AI) systems have become increasingly common and the trend will continue. Examples of AI systems include autonomous vehicles (AV), computer vision, natural language processing, and AI medical experts. To allow for safe and effective deployment of AI systems, the reliability of such systems needs to be assessed. Traditionally, reliability assessment is based on reliability test data and the subsequent statistical modeling and analysis. The availability of reliability data for AI systems, however, is limited because such data are typically sensitive and proprietary. The California Department of Motor Vehicles (DMV) oversees and regulates an AV testing program, in which many AV manufacturers are conducting AV road tests. Manufacturers participating in the program are required to report recurrent disengagement events to California DMV. This information is being made available to the public. In this paper, we use recurrent disengagement events as a representation of the reliability of the AI system in AV, and propose a statistical framework for modeling and analyzing the recurrent events data from AV driving tests. We use traditional parametric models in software reliability and propose a new nonparametric model based on monotonic splines to describe the event process and to estimate the cumulative baseline intensity function of the event process. We develop inference procedures for selecting the best models, quantifying uncertainty, and testing heterogeneity in the event process. We then analyze the recurrent events data from four AV manufacturers, and make inferences on the reliability of the AI systems in AV. We also describe how the proposed analysis can be applied to assess the reliability of other AI systems. This paper has online supplementary materials.

Key Words: Disengagement Events; Fractional Random Weight Bootstrap; Gompertz Model; Monotonic Splines; Software Reliability; Self-driving Cars.

1 Introduction

1.1 The Problem

With the rapid development of new technology, artificial intelligence (AI) systems are emerging in many areas. Typical applications of AI systems include autonomous vehicles (AV), computer vision, speech recognition, and AI medical experts. The reliability and safety of AI systems need to be assessed before massive deployment in the field. For example, the reliability of AV needs to be demonstrated so that people can use them with confidence. Traditionally, reliability assessment of products and systems is based on reliability test data collected from the laboratory and the field. Reliability information is then extracted from statistical modeling and analysis of the data.

Commonly used data types for reliability analysis are time-to-event data, degradation data, and recurrent events data. Reliability data collected by manufacturers are highly sensitive and are usually not publicly available. In the area of AV, however, the California Department of Motor Vehicles (DMV) launched an AV driving program. More details of the study are given in Section 2. Many AV manufacturers are participating in the program and so are conducting their AV road tests in California. As part of their participation, manufacturers are required to report disengagement events and mileage information to the California DMV. The reported events are available for public access. Because of the availability of these recurrent events data, we focus on the reliability analysis of the AI systems in AV units in this paper.

A disengagement event happens when the AI system and/or the backup driver determines that the driver needs to take over the driving. The recurrence rates of disengagement events can be used as a proxy for the reliability of the AI system in AV. A lower occurrence rate (event rate) of disengagement events would indicate a more reliable AI system in the AV. In the reliability literature, parametric forms have been used to describe the event rate through a nonhomogeneous Poisson process (NHPP) model. In practice, some specific questions arise in the analysis of the recurrent disengagement events data,

- How to model the event process, and what kind of parametric forms should be used?
- Does the parametric form provide an adequate fit to the data, and are there any other flexible forms for modeling?
- Is there any population heterogeneity in the event processes from multiple test units?

We develop a statistical framework for modeling and analyzing the recurrent events data from AV driving tests to answer these practical questions. Specifically, we apply the NHPP model to describe the disengagement event process with adjustment for the time-varying mileage data using parametric models that are used to describe cumulative intensity functions in software reliability applications. We also propose a new nonparametric model based on

monotonic splines to describe event processes. The spline model is flexible enough to fit various patterns in the event process and can also be used to assess whether the fully-parametric model provides an adequate fit. We develop inference procedures for selecting the best models and quantifying uncertainty using the fractional-random-weight bootstrap. The parametric models and spline models are complementary tools. In addition, we use the gamma frailty model to quantify and assess heterogeneity in an event process.

From the California driving study, we use data from four manufacturers that have been conducting extensive AV driving tests in California. We apply the developed methods to analyze the recurrent events data from the four AV manufacturers. Based on the modeling, we make inferences on the reliability of the AI systems in AV, and summarize interesting findings on the reliability of the AI systems in AV. Although our analysis focuses on the AI systems in AV, the statistical analysis can also be applied to assess the reliability of other AI systems.

1.2 Literature Review

There is only a small amount of literature on reliability analysis of AI systems. Amodei et al. (2016) provided a general discussion about AI safety and outlined five concrete areas for AI safety research. Bosio et al. (2019) conducted a reliability analysis of deep convolutional neural networks (CNN) developed for automotive applications using fault injections. Goldstein et al. (2020) investigated the impact of transient faults on the reliability of compressed deep CNN. Zhao et al. (2020) proposed a safety framework based on Bayesian inference for critical systems using deep learning models. Alshemali and Kalita (2020) provided a review of methods for improving the reliability of natural language processing. Due to the limited availability of test data, statistical reliability analysis of AI reliability/safety is in an emerging stage.

As an example of the modeling of the reliability and safety of AV, Kalra and Paddock (2016) used a statistical hypothesis testing approach to determine the needed miles of driving to demonstrate AV safety. Åsljung, Nilsson, and Fredriksson (2017) used extreme value theory to model the safety of AV. Michelmore et al. (2019) designed a statistical framework to evaluate the safety of deep neural network controllers and assessed the safety of AV. Burton et al. (2020) provided a multi-disciplinary perspective on the safety of AV from engineering, ethics, and law aspects. Most existing modeling frameworks, however, do not involve large scale field-testing data. Such data are essential for reliability assessment.

The California driving test data provide unique opportunities for data analysis. Regarding the analysis of the California driving data, Dixit, Chand, and Nair (2016) and Favarò, Eurich, and Nader (2018) analyzed the causes of disengagement events using the California driving test data up to 2017 and showed the relationship between disengagement events per mile

and cumulative miles. Lv et al. (2018) performed a descriptive analysis of the causes of disengagement events using the California driving test data from 2014 to 2015, and concluded that software issues and limitations were the most common reasons for disengagement events. Banerjee et al. (2018) used linear regression models to describe the relationship between disengagements per mile and cumulative miles using the data from 2016 to 2017. Merkel (2018) analyzed the California driving data from 2015 to 2017 using aggregated counts data and least-squares fit. Zhao et al. (2019) proposed a general conservative Bayesian inference method to estimate the rate of events (crashes and fatalities) and illustrated it with the California driving test data. Boggs, Wali, and Khattak (2020) did an exploratory analysis for AV crashes from the California driving study. So far, there is no comprehensive statistical treatment for the analysis of AI reliability and especially for AV reliability. Starting in 2018, the exact disengagement event times can be extracted from the California DMV report, which makes it possible to apply recurrent events modeling techniques.

In the reliability literature, NHPP models are widely used to analyze recurrent events. Zuo, Meeker, and Wu (2008), and Hong, Li, and Osborn (2015) analyzed recurrent events data with window-observations, which has similar data types with the disengagement events data from the California driving study. Parametric models, such as the Musa-Okumoto model (e.g., Musa and Okumoto 1984) and the Gompertz model (e.g., Huang, Lyu, and Kuo 2003), are commonly used in software reliability applications (e.g., Wood 1996). Ehrlich et al. (1998) used accelerated testing methods to study software reliability. Burke, Jones, and Noufaily (2020) proposed flexible parametric models for time-to-event data analysis. Useful reference books for reliability data analysis for researchers in the AI reliability area include Lawless (2003), and Meeker, Escobar, and Pascual (2021). Overall, parametric models are common in analyzing recurrent events data in the context of reliability studies.

We propose to use monotonic splines (Ramsay 1988, and Meyer 2008) as a nonparametric method to model the event process. Although monotonic splines are used in some degradation settings (Xie et al. 2018), the application of monotonic splines to recurrent events in reliability is new. To model population heterogeneity, the gamma frailty model is used. An early use of the gamma frailty model in reliability is found in Lawless (1995) with additional work and applications in Cook and Lawless (2007). More recently, Shan, Hong, and Meeker (2020) used the gamma frailty model to describe seasonal warrant return data. Duchateau and Janssen (2008) provide a comprehensive review of frailty models.

Due to the complicated structure of the window-observed recurrent events data in our study, we use fractional-random-weight bootstrap as a convenient way to generate bootstrap samples for statistical inference. The idea of fractional-random-weight bootstrap is introduced in Rubin (1981), and some theoretical properties are shown in Jin, Ying, and Wei (2001). Xu et al. (2020) demonstrated the use of fractional-random-weight bootstrap in many complex

applications in reliability, survival analysis, and regression. Simultaneous confidence bands (SCB) can be used to assess if a parametric model is adequate for fitting the data. Hong, Escobar, and Meeker (2010) showed that an SCB could be obtained from a simultaneous confidence region (SCR) for parameters. However, in the context of bootstrap, it is not straightforward to construct SCR for parameter estimators with multiple dimensions. Hence, we use the idea of the equal-precision band in Nair (1984) and use bootstrap samples to calibrate pointwise confidence intervals to provide SCB to quantify statistical uncertainty.

In summary, we provide a general analytic framework by integrating existing methods and proposing new methods for reliability analysis of data from an AI study. The parametric and nonparametric models, and the statistical interval and testing procedures will be useful tools for practitioners working in the area of AI reliability.

1.3 Overview

The rest of the paper is organized as follows. Section 2 describes the California autonomous vehicle driving study and introduces the data. Section 3 describes the parametric model, the spline model, and the gamma frailty model that are used to describe the recurrent disengagement events data. Section 4 describes the parameter estimation procedures and the inference procedures for various models. Section 5 conducts a simulation study to the statistical performance of the estimation procedures. Section 6 conducts the data analysis, summarizes interesting findings, and compares with existing methods. Section 7 contains some concluding remarks and areas for future research.

2 The California Autonomous Vehicles Driving Study

2.1 The Study

This paper presents reliability modeling and analysis of autonomous vehicles (AV) using data from the California Department of Motor Vehicles (DMV) autonomous vehicle tester program, which has been in operation since 2014. The tester program allows manufacturers to test AV on California public roads with a human in the driver seat who can take control of the vehicle if necessary. Up to July 1, 2020, 62 manufacturers had been permitted to perform AV drive testing. Manufacturers are required to report disengagement events annually, and collision of AV within 10 days of the accident. Before December 1, 2017, only the aggregated number of disengagement events per month was reported. Since then, the exact date of the event is now reported. Thus, we focus on the analysis of the data after December 1, 2017.

Because it can be difficult to determine responsibility in collisions, a disengagement event

is typically used as an alternative to determining unsafe auto-driving in the literature (e.g., Merkel 2018). During the period from December 1, 2017, to November 30, 2019, 34 manufacturers reported disengagement events. We use the data from disengagement events reported by Waymo, Cruise, Pony AI, and Zoox because these four manufacturers performed extensive on-road testing during this time period.

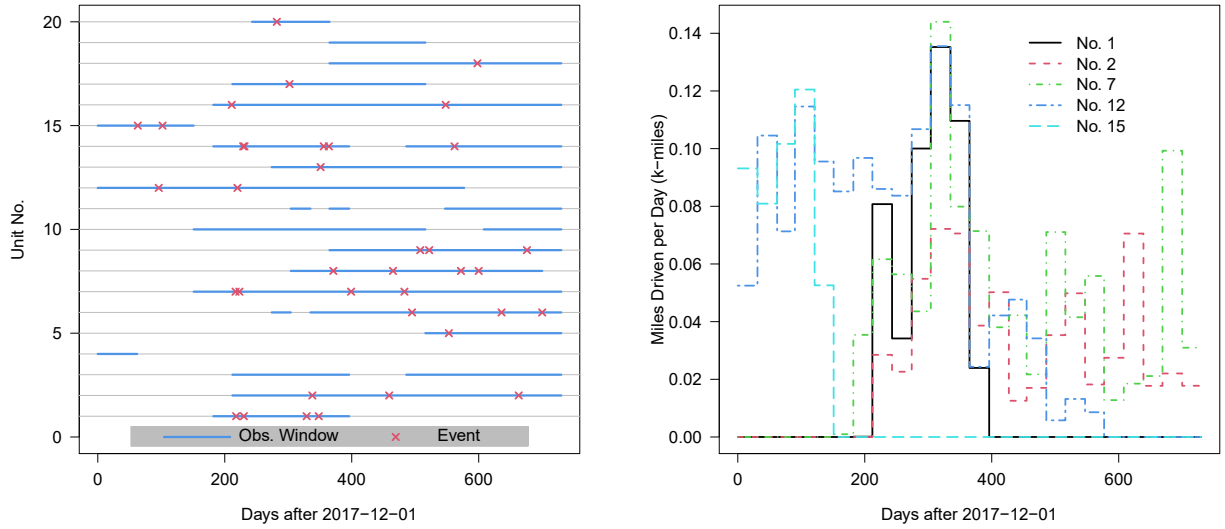
Here we provide a brief introduction to those four manufacturers. Waymo began as the Google Self-Driving Car Project in 2009, testing autonomous vehicles on public roads across six states in the United States. Waymo joined the California tester program in 2015. Cruise is an autonomous vehicle company founded in 2013. It joined the California tester program in 2016 and tested AV in the urban environment of San Francisco. Pony AI was founded in 2016, developing autonomous driving technology globally. Pony AI started AV testing on California public roads in June 2017. Zoox is an autonomous vehicle company founded in 2013. They joined the California tester program in 2017 and tested AV in downtown San Francisco.

2.2 Disengagement Events Data

The California DMV requires manufacturers to report when their vehicles disengaged from autonomous mode during tests. A disengagement event occurs when there is an autonomous technology failure, or when a situation requires the test driver to take manual control of the vehicle to operate safely. Disengagement events can be initialized by a warning from the autonomous vehicle system, or by test drivers as the driver thinks it is not safe to continue auto driving. Disengagement reports are provided annually, and include the ID number of vehicles in testing, location and date of disengagement events, description of cause of disengagement, monthly autonomous mileage of each testing vehicle, and annual total autonomous miles of each testing vehicle.

The study period for this paper is from December 1, 2017, to November 30, 2019, which is a 24 month or 2 year study period. The data for the period from December 1, 2018, to November 30, 2019, are available in csv format from the California DMV website, while the data for the period from December 1, 2017, to November 30, 2018, are available in pdf format that need to be manually converted to csv format.

After data cleaning, an event time is computed as the number of days since the starting date. Because only monthly mileage was reported, the monthly mileage is divided by the number of days in that month to obtain the daily mileage. Under the approximation, the daily mileage is constant over each month. The unit for the mileage is thousands of miles (k-miles). Figure 1 shows a subset of the recurrent events data and mileage data from manufacturer Waymo. Figure 1(a) shows the recurrent events data with the crosses representing the event times and the thicker horizontal segments showing the active months (i.e., events can only



(a) Recurrent Events

(b) Thousands of Miles per day

Figure 1: Visualization of a subset of the recurrent events data and mileage data from manufacturer Waymo. (a) The recurrent events data with the crosses showing the event times and the thicker horizontal segments showing the active months. (b) A plot of the mileage driven per day as a function of time for five representative AV. Note that the miles driven are in the units of thousands of miles (k-miles).

be recorded within the observation windows). Figure 1(b) plots the mileage as a function of time for five representative units. Table 1 shows a summary of the recurrent events data and mileage data from the four manufacturers. We can see that Waymo and Cruise have driven more than 1 million miles and the disengagement event rate is around 0.1 events per k-miles during the 24 month testing period. Pony AI and Zoox have smaller amounts of driving miles, and the event rates are around 0.2 events per k-miles and 0.6 events per k-miles, respectively.

2.3 Notation for Data

The number of AV testing units (vehicles) in a fleet is denoted by n. The total observation time is $\tau = 730$ days (i.e., two years). Let $t_{ij}, i = 1, \ldots, n, j = 1, \ldots, n_i$ be event time j for unit i. Here t_{ij} records the number of days since December 1, 2017, and n_i is the number of events for unit i. Note that it is possible that $n_i = 0$, indicating that there were no events observed for unit i.

Let $x_i(t), 0 < t \leq \tau$, be the mileage driven for unit i at time (day) t. The unit of $x_i(t)$ is k-miles. The daily average of monthly mileage was used for $x_i(t)$. Thus, $x_i(t)$ is a step

Table 1: Summary of the recurrent events data and mileage data from the four manufacturers over the 24 month study period.

Manufacturer	No. of Vehicles	Active Months	Active Months per Vehicle	No. of Events	Total in k-miles	No. of Events per k-miles
Waymo	123	1550	12.602	224	2710.136	0.083
Cruise	304	2079	6.839	154	1278.661	0.120
Pony AI	23	179	7.783	43	190.871	0.225
Zoox	32	280	8.750	58	97.780	0.593

function, which can be represented as,

$$x_i(t) = \sum_{l=1}^{n_\tau} x_{il} \mathbb{1}(\tau_{l-1} < t \leq \tau_l). \tag{1}$$

Here, $n_\tau = 24$ is the number of months in the follow-up period, x_{il} is the daily mileage for unit i during month l, τ_l is the ending day since the start of the study for month l, and $\mathbb{1}(\cdot)$ is an indicator function. Let $\boldsymbol{x}_i(t) = \{x_i(s) : 0 < s \leq t\}$ be the history for the mileage driven for unit i.

3 Statistical Models

3.1 The Nonhomogeneous Poisson Process

Recurrent events processes are commonly modeled by a nonhomogeneous Poisson process (NHPP). The event intensity function for unit i is modeled as,

$$\lambda_i[t; x_i(t), \boldsymbol{\theta}] = \lambda_0(t; \boldsymbol{\theta}) x_i(t). \tag{2}$$

Here, $\lambda_0(t; \boldsymbol{\theta}) = \lambda_0(t)$ is the baseline intensity function (BIF) with parameter vector $\boldsymbol{\theta}$. Because $x_i(t)$ is the mileage driven, $\lambda_i[t; x_i(t), \boldsymbol{\theta}]$ is the mileage-adjusted event intensity. The BIF can be interpreted as the event rate per k-miles at time t when $x_i(t) = 1$. The baseline cumulative intensity function (CBIF) is,

$$\Lambda_0(t; \boldsymbol{\theta}) = \Lambda_0(t) = \int_0^t \lambda_0(s; \boldsymbol{\theta}) ds. \tag{3}$$

Note that $\Lambda_0(0; \boldsymbol{\theta}) = 0$ and $\Lambda_0(t; \boldsymbol{\theta})$ is a non-decreasing function of t. The CBIF $\Lambda_0(t)$ can be interpreted as the expected number of events from time 0 to t when $x(t) = 1$ for all t. The cumulative intensity function (CIF) for unit i is,

$$\Lambda_i[t; x_i(t), \boldsymbol{\theta}] = \int_0^t \lambda_0(s; \boldsymbol{\theta}) x_i(s) ds. \tag{4}$$

Table 2: List of commonly used parametric models and their BIFs, CBIFs, and parameters.

Model	CBIF $\Lambda_0(t;\boldsymbol{\theta})$	BIF $\lambda_0(t;\boldsymbol{\theta})$	Parameters
Musa-Okumoto	$\theta_1^{-1}\log(1+\theta_2\theta_1 t)$	$\theta_2(1+\theta_2\theta_1 t)^{-1}$	$\boldsymbol{\theta}=(\theta_1,\theta_2)'$ $\theta_1>0,\theta_2>0$
Gompertz	$\theta_1\theta_3^{\theta_2^t}-\theta_1\theta_3$	$\theta_1\theta_2^t\theta_3^{\theta_2^t}\log(\theta_2)\log(\theta_3)$	$\boldsymbol{\theta}=(\theta_1,\theta_2,\theta_3)'$ $\theta_1>0,0<\theta_2,\theta_3<1$
Weibull	$\theta_1[1-\exp(-\theta_2 t^{\theta_3})]$	$\theta_1\theta_2\theta_3 t^{(\theta_3-1)}\exp(-\theta_2 t^{\theta_3})$	$\boldsymbol{\theta}=(\theta_1,\theta_2,\theta_3)'$ $\theta_1>0,\theta_2>0,\theta_3>0$

In software reliability, the CBIF and the BIF are used as reliability metrics when recurrent events can be collected from repairable systems (e.g., Wood 1996). The trends in the BIF can indicate the evolution of the reliability in the underlying AV system. For example, improvement of the autonomous technology in AV can lead to a decreasing trend in the BIF (i.e., CBIF increasing at a decreasing rate).

Typically in software reliability, one specifies a parametric form for the CBIF $\Lambda_0(t;\boldsymbol{\theta})$. Note that the BIF can be obtained by taking the derivative of CBIF with respect to t. That is, $\lambda_0(t;\boldsymbol{\theta})=d\Lambda_0(t;\boldsymbol{\theta})/dt$. The commonly used models for $\Lambda_0(t;\boldsymbol{\theta})$ are Musa-Okumoto, Gompertz, and the Weibull models (e.g., Merkel 2018). Table 2 lists their BIFs, CBIFs, and parameters. Note that the Weibull CBIF is similar to the Weibull cumulative distribution function (cdf) but with an asymptote θ_1 as t goes to ∞.

3.2 Spline Models

Although parametric models can fit certain trends in the event process, they may not be flexible enough to describe the event process for AV testing, as the evolution of the AI technology in an AV system can be complicated, which motivates us to propose the I-spline model for describing the CBIF. In the I-spline model, the CBIF is represented as a linear combination of spline bases. That is,

$$\Lambda_0(t;\boldsymbol{\theta})=\sum_{l=1}^{n_s}\beta_l\gamma_l(t),\quad \beta_l\geq 0,\, l=1,\ldots,n_s, \tag{5}$$

provides a nonparametric method to describe the CBIF. Here $\boldsymbol{\theta}=(\beta_1,\ldots,\beta_{n_s})'$ is the vector for the spline coefficients, $\gamma_l(t)$'s are the spline bases, and n_s is the number of spline bases. The BIF can be obtained by taking a derivative with respect to t. That is,

$$\lambda_0(t;\boldsymbol{\theta})=\frac{d\Lambda_0(t;\boldsymbol{\theta})}{dt}=\sum_{l=1}^{n_s}\beta_l\frac{d\gamma_l(t)}{dt}. \tag{6}$$

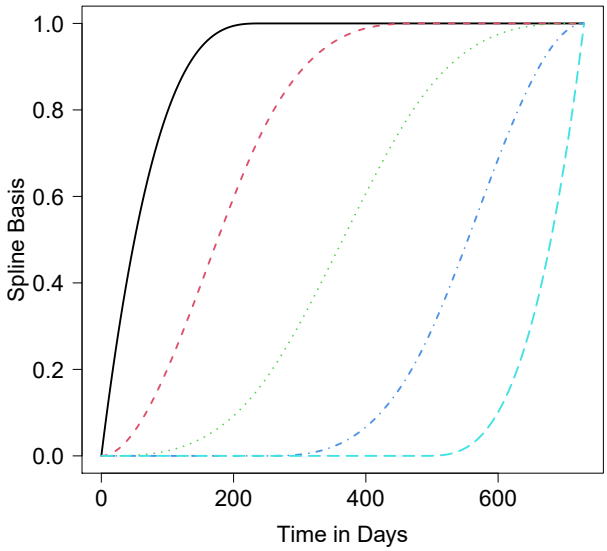


Figure 2: Examples of I-spline basis functions.

Because of the constraints that $\Lambda_0(0;\boldsymbol{\theta}) = 0$ and that $\Lambda_0(t;\boldsymbol{\theta})$ is a non-decreasing function of t, some special considerations are needed in the I-spline model. We use the I-splines, described in Ramsay (1988). Figure 2 shows examples of I-spline basis functions (i.e., $\gamma_l(t)$). We can see that each spline basis takes value zero at $t = 0$ and is monotonically increasing. By taking non-negative coefficients (i.e., $\beta_l \geq 0$), a non-decreasing $\Lambda_0(t;\boldsymbol{\theta})$ is obtained.

A brief introduction on the construction of I-spline basis is as follows. The boundary knots are 0 and τ. The b interior knots are denoted by $t_{h+1}, \ldots, t_{h+b}$ for splines of order h. The complete sequence for the knots are denoted by $0 = t_1 = \cdots = t_h < t_{h+1} < \cdots < t_{h+b} < t_{h+b+1} = \cdots = t_{2h+b} = \tau$. The total number of spline bases is $n_s = h+b$. I-splines are obtained by integrating the M-splines; that is,

$$I_q^{(h)}(t) = \int_0^t M_q^{(h)}(u)du, q = 1, \ldots, b+h, \quad t \in [0, \tau],$$

and the M-spline bases of order h are defined recursively. The M-splines of order 1 is

$$M_q^{(1)}(t) = \mathbb{1}(t_q \leq z < t_{q+1})(t_{q+1} - t_q)^{-1},$$

for $q = 1, \cdots, b+1$. The M-splines of order h are obtained as

$$M_q^{(h)}(t) = \frac{h[(t-t_q)M_q^{(h-1)}(t) + (t_{q+h} - t)M_{q+1}^{(h-1)}(t)]}{(h-1)(t_{q+h} - t_q)}\mathbb{1}(t_q \leq t < t_{q+h}),$$

for $q = 1, \cdots, b+h$.

3.3 Modeling Heterogeneity

It is maybe possible that some AV units are more likely to generate more events even after accounting for the mileage driven, resulting in heterogeneity in the event process. Similar to the approach in Chapter 3 of Cook and Lawless (2007), a frailty term can be added to the parametric intensity function to model this extra heterogeneity. The gamma frailty model is,

$$\lambda_i[t; u_i, \boldsymbol{x}_i(t), \boldsymbol{\theta}] = u_i \lambda_0(t; \boldsymbol{\theta}) x_i(t). \tag{7}$$

Here, the frailty term u_i is a random variable that has a gamma distribution with mean one and variance ϕ. The probability density function (pdf) of u_i is,

$$f(u_i) = \frac{1}{\Gamma(1/\phi)\phi^{1/\phi}} u_i^{(1/\phi - 1)} \exp(-u_i/\phi).$$

The gamma frailty model is popular in reliability and survival applications because there is a closed-form expression for the marginal likelihood for recurrent events data based on the model in (7).

4 Model Estimation and Inference

This section presents parameter estimation procedures for the parametric and I-spline models, and the corresponding statistical inference procedures. This section also contains parameter estimation and hypothesis testing for the gamma frailty model.

4.1 Parameter Estimation

We use the maximum likelihood (ML) methods for parameter estimation. The likelihood function is,

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} \left\{ \prod_{j=1}^{n_i} \lambda_i[t_{ij}; x_i(t_{ij}), \boldsymbol{\theta}] \right\} \times \exp\{-\Lambda_i[\tau; \boldsymbol{x}_i(\tau), \boldsymbol{\theta}]\}, \tag{8}$$

with the convention that $\prod_{j=1}^{0}(\cdot) = 1$. Here, the intensity function and the CIF are defined in (2) and (4), respectively. The log-likelihood function is obtained by taking the logarithm of the $L(\boldsymbol{\theta})$ in (8). That is,

$$\begin{aligned} l(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \left(\sum_{j=1}^{n_i} \{\log[x_i(t_{ij})] + \log[\lambda_0(t_{ij}; \boldsymbol{\theta})]\} \right) - \Lambda_i[\tau; \boldsymbol{x}_i(\tau), \boldsymbol{\theta}] \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n_i} \{\log[x_i(t_{ij})] + \log[\lambda_0(t_{ij}; \boldsymbol{\theta})]\} - \sum_{i=1}^{n} \sum_{l=1}^{n_\tau} \{x_{il} \cdot [\Lambda_0(\tau_l; \boldsymbol{\theta}) - \Lambda_0(\tau_{l-1}; \boldsymbol{\theta})]\}, \end{aligned} \tag{9}$$

with the convention that $\sum_{j=1}^{0}(\cdot) = 0$. The last step in (9) is obtained by substituting the expression for $x_i(t)$ in (1). Note that we need enough events in the data so that the estimated intensity function will not be zero and so that not all of the spline coefficients will be zero. For example, using a parametric model with three parameters, at least three distinct event times are needed to estimate the parameters from the data.

For parametric models, the functional forms of $\Lambda_0(t)$ and $\lambda_0(t)$ in Table 2 can be substituted into (9) to evaluate the log-likelihood function. The ML estimate of $\boldsymbol{\theta}$, denoted by $\widehat{\boldsymbol{\theta}}$, can be obtained by finding the value of $\boldsymbol{\theta}$ that maximizes $l(\boldsymbol{\theta})$. For parametric models, the length of $\boldsymbol{\theta}$ is typically 2 or 3. We used the R `optim()` function with the "Nelder-Mead" option to do the optimization.

For the I-spline model, the functional forms of $\Lambda_0(t)$ and $\lambda_0(t)$ in (5) and (6), respectively, can be substituted into (9) to evaluate the log-likelihood function. To estimate the parameter $\boldsymbol{\theta}$ for the I-spline model, one needs to specify the locations of the knots and the number of knots, and address the non-negativity constraints on $\boldsymbol{\theta}$ as indicated in (5).

We use I-splines of order 3. Note that an I-spline with order 3 is an integral of an M-spline with order 3, and is differentiable up to the fourth order, which is generally smooth enough to fit the CBIF. The boundary knots are set to be 0 on the left side and $\tau = 730$ on the right side. The interior knots are set to be equally spaced sample quantiles of the observed event times. For example, if three interior knots are needed, the interior knots are set to the 0.25, 0.5, and 0.75 sample quantiles of the observed event times. After setting the knot locations, the spline bases can be computed.

To select the best number of knots, we use the Akaike information criterion (AIC), which is computed as

$$\text{AIC} = -2l(\boldsymbol{\theta}) + 2df.$$

Here, df is the number of degrees of freedom for the model. For the I-spline model, df is the number of non-zero coefficients. The number of spline coefficients is the order of the M-splines plus the number of interior knots. In estimation, we usually try a large enough range of values for the number of knots. Typically, the AIC value decreases when the number of knots increases at the beginning, and then starts to increase after the number of knots is larger than some value. The maximum number of interior knots is selected to be larger than the change point. In practice, we found the change point is related to the number of events in the data. For example, for Waymo and Cruise, we tried the number of interior knots up to 17, and the number of events recorded for Waymo and Cruise are 224 and 154, respectively. For Pony AI, the number of interior knots is up to 7, and the number of events is 43. For Zoox, the number of interior knots is up to 11, and the number of events is 58.

To address the non-negativity constraints on $\boldsymbol{\theta}$, we use the "L-BFGS-B" option in the

R `optim()` function. The "L-BFGS-B" algorithm allows users to specify an interval for the variable to be optimized. We set the interval to be $[0, \infty)$ for the spline coefficients. Using the "L-BFGS-B" algorithm, some of the elements of the estimates $\widehat{\boldsymbol{\theta}}$ could be set to zero, and the corresponding spline basis then does not contribute to the cumulative intensity function.

4.2 Statistical Inference

For inference based on parametric models, the normal approximation based on large sample theory can be used. Because the inference for parametric models is relatively straightforward, this section focuses on the inference for the I-spline model. For the I-spline model, the normal approximation is inappropriate because the parameter estimates can occur at the boundary of the parameter space (i.e., $\widehat{\beta}_l$ can be zero). Thus, we use the fractional-random-weight bootstrap (e.g., Xu et al. 2020). Unlike other bootstrap methods, the fractional-random-weight bootstrap provides a convenient way to quantify uncertainty for data that has a complex structure. For our case, we need to handle recurrent events with window observations and time-varying mileage information, which complicate the structure of the data.

The implementation of the fractional-random-weight bootstrap is straightforward. The log-likelihood function in (9) is re-weighted as

$$\begin{aligned} l^*(\boldsymbol{\theta}) = & \sum_{i=1}^{n} \sum_{j=1}^{n_i} w_i \left\{ \log[x_i(t_{ij})] + \log[\lambda_0(t_{ij}; \boldsymbol{\theta})] \right\} \\ & - \sum_{i=1}^{n} \sum_{l=1}^{n_\tau} w_i \left\{ x_{il} \cdot [\Lambda_0(\tau_l; \boldsymbol{\theta}) - \Lambda_0(\tau_{l-1}; \boldsymbol{\theta})] \right\}, \end{aligned} \tag{10}$$

where the random weights are independently generated from an exponential distribution with mean one. The bootstrap algorithm for generating bootstrap versions of the estimate of CBIF $\widehat{\Lambda}_0^*(t)$ is summarized as follows.

Algorithm 1: Bootstrap Algorithm for Generating $\widehat{\Lambda}_0^*(t)$.

1. Generate random weights $w_i, i = 1, \ldots, n$, independently from an exponential distribution with mean one.
2. Construct the re-weighted log-likelihood function as in (10).
3. Use AIC to select the best number of knots based on the log-likelihood function in (10).
4. Based on the best number of knots chosen in step 3, the corresponding spline bases, and estimated coefficients, one can compute estimates for CBIF, denoted by $\widehat{\Lambda}_0^*(t)$.
5. Repeat steps 1 to 4 to obtain B copies of $\widehat{\Lambda}_0^*(t)$, denoted by $\widehat{\Lambda}_0^{*b}(t), b = 1, \ldots, B$.

Based on the bootstrap estimates $\widehat{\Lambda}_0^{*b}(t), b = 1, \ldots, B$, one can construct an approximate $100(1-\alpha_p)\%$ pointwise confidence interval (PCI) for $\Lambda_0(t)$ for a given t,

$$\left[\widehat{\Lambda}_0^{*([B\alpha_p/2])}(t),\ \widehat{\Lambda}_0^{*([B(1-\alpha_p/2)])}(t)\right].$$

Here, $\widehat{\Lambda}_0^{*(b)}(t)$ is the ordered version of $\widehat{\Lambda}_0^{*b}(t)$, and $[\,\cdot\,]$ is the rounding function.

Based on the I-spline model, one can construct a simultaneous confidence band (SCB) for the CBIF, which can be used to assess the adequacy of the parametric models in fitting the data. An approximate $100(1-\alpha)\%$ SCB for $\Lambda_0(t), t_L \le t \le t_U$, is

$$\left[\underset{\sim}{\Lambda}_0(t),\ \widetilde{\Lambda}_0(t)\right], \quad t_L \le t \le t_U, \tag{11}$$

where t_L and t_U are boundaries to be specified. If an estimated parametric model is contained in the SCB in (11) the parametric model is statistically consistent with the data (i.e., there is no statistical evidence to reject the parametric model). Thus, the SCB constructed by the I-spline model provides a tool to check if a parametric model is adequate.

The $100(1-\alpha_p)\%$ PCIs provide a structure for computing SCBs for $\Lambda_0(t)$ for all in the $t \in [t_L,\, t_U]$ period. We use bootstrap samples to calibrate the PCIs so that it can approximately provide the nominal $100(1-\alpha)\%$ coverage probability (CP), similar to the idea of the equal-precision SCB for a cdf in Nair (1984). The CP can be estimated as

$$\mathrm{CP}(\alpha_p) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}\left(\widehat{\Lambda}_0^{*([B\alpha_p/2])}(t) \le \widehat{\Lambda}_0^{*b}(t) \le \widehat{\Lambda}_0^{*([B(1-\alpha_p/2)])}(t),\ \text{for all } t \in [t_L,\, t_U]\right).$$

By setting $\mathrm{CP}(\alpha_p) = 1-\alpha$, one can find the solution to be α_c. Thus, the SCB in (11) can be computed as

$$\left[\widehat{\Lambda}_0^{*([B\alpha_c/2])}(t),\ \widehat{\Lambda}_0^{*([B(1-\alpha_c/2)])}(t)\right], \quad t_L \le t \le t_U,$$

which is time-efficient because the bootstrap samples have already been obtained.

4.3 The Frailty Model

To estimate the gamma frailty in (7), one needs to calculate the marginal likelihood function. The marginal likelihood function is,

$$\begin{aligned} L(\boldsymbol{\theta}, \phi) &= \prod_{i=1}^{n} \int_0^{\infty} \left\{\prod_{j=1}^{n_i} u_i \lambda_i[t_{ij}; x_i(t_{ij}), \boldsymbol{\theta}]\right\} \times \exp\{-u_i \Lambda_i[\tau; \boldsymbol{x}_i(\tau), \boldsymbol{\theta}]\} f(u_i) du_i \qquad (12) \\ &= \prod_{i=1}^{n} \left\{\prod_{j=1}^{n_i} \lambda_i[t_{ij}; x_i(t_{ij}), \boldsymbol{\theta}]\right\} \times \int_0^{\infty} u_i^{n_i} \exp(-u_i c_i) f(u_i) du_i \\ &= \prod_{i=1}^{n} \left\{\prod_{j=1}^{n_i} \lambda_i[t_{ij}; x_i(t_{ij}), \boldsymbol{\theta}]\right\} \times \frac{\phi^{-1/\phi}\Gamma(n_i + 1/\phi)}{\Gamma(1/\phi)(c_i + 1/\phi)^{(n_i+1/\phi)}}, \end{aligned}$$

where $c_i = \sum_{l=1}^{n_\tau} x_{il} \cdot [\Lambda_0(\tau_l; \boldsymbol{\theta}) - \Lambda_0(\tau_{l-1}; \boldsymbol{\theta})]$. The ML estimates of $\boldsymbol{\theta}$ and ϕ are obtained by finding the values that maximize the log-likelihood function, $\log[L(\boldsymbol{\theta}, \phi)]$. We again use the R `optim()` function with the "Nelder-Mead" option to do the optimization.

To check for population heterogeneity in the event process, one can use a likelihood ratio test. The test statistic is constructed as,

$$-2\{l(\widehat{\boldsymbol{\theta}}) - \log[L(\widehat{\boldsymbol{\theta}}, \widehat{\phi})]\}, \tag{13}$$

which has a χ^2_1 distribution under the null hypothesis that $\phi = 0$. Here, $l(\widehat{\boldsymbol{\theta}})$ is obtained by evaluating (9) at $\widehat{\boldsymbol{\theta}}$. The null hypothesis is rejected if the test statistic is larger than $\chi^2_{1,(1-\alpha)}$, indicating evidence for population heterogeneity.

5 Simulation Study

The purpose of the simulation study is to show the properties of the ML estimator for the I-spline model, to check the CP of the SCB procedure based on the I-spline model, and see if the SCB can correctly accept or reject a particular parametric model.

5.1 Setting

In the simulation study, the I-spline model is taken to be the true underlying model. The spline bases are shown in Figure 2. We consider three scenarios. Figure 3 shows the true CBIFs used in the three simulation scenarios. To simplify the setting, we only consider the Gompertz model.

- Scenario 1: we choose $\boldsymbol{\theta} = (6, 16, 23, 11, 4)'$ as the coefficients. Using the spline bases in Figure 2, the true CBIF is shown as the black/solid line in Figure 3. The Gompertz model fits this CBIF perfectly.

- Scenario 2: we choose $\boldsymbol{\theta} = (8, 12, 28, 0, 12)'$ as the coefficients. The corresponding true CBIF is shown as the red/dash line in Figure 3. The Gompertz model fits this CBIF well except for the late stage.

- Scenario 3: we choose $\boldsymbol{\theta} = (5, 25, 0, 30, 0)'$ as the coefficients. The corresponding true CBIF is shown as the green/dot-dash line in Figure 3. The Gompertz model does not fit this CBIF well due to various changes in the slope over time.

The number of bootstrap samples is $B = 5000$ and the number of simulated datasets (i.e., repeats) is $N = 1000$. The mileage driven history is sampled with replacement from the

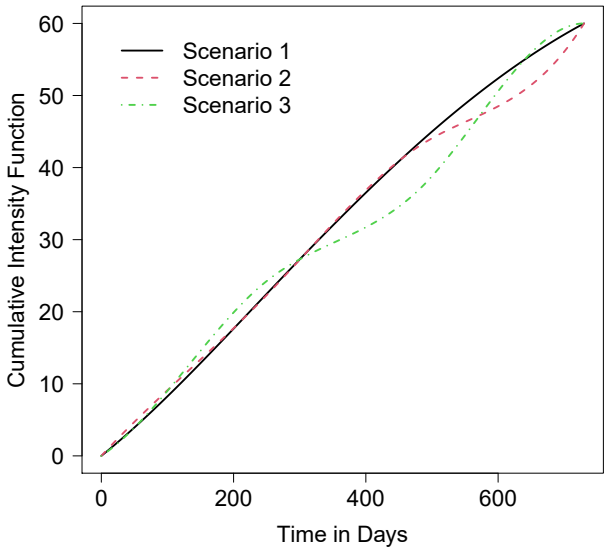


Figure 3: Plot of the true CBIFs used in the three simulation scenarios.

historical Waymo data. The average number of events per unit is around 1.8. The sample sizes (i.e., the number of AV units) considered in the simulation are 50, 100, 200, 500, and 1000.

We evaluate the relative root mean squared errors (RMSE) for the CBIF estimator, the CP for the SCB, and the acceptance probability for the parametric model. For each scenario, we considered 12 spline models, in which the number of interior knots varies from 1 to 12. Then, we used AIC to select the best number of interior knots as the best spline model to check the CP of SCB procedure.

In particular, the relative RMSE (RelRMSE) is computed as

$$\text{RelRMSE} = \frac{\left\{\sum_{l=1}^{N}\left[\widehat{\Lambda}_{0l}(t) - \Lambda_0(t)\right]^2 / N\right\}^{1/2}}{\Lambda_0(t)},$$

where $\widehat{\Lambda}_{0l}(t)$ is the estimated CBIF using the I-spline model based on the lth simulated dataset, and $\Lambda_0(t)$ is the true CBIF. We use the relative RMSE to remove the scale effect of $\Lambda_0(t)$. That is, RMSE tends to be large if $\Lambda_0(t)$ is large at a particular t. The CP is estimated by the proportion that the SCB constructed by using the I-spline model captures the true CBIF. The acceptance probability is estimated by the proportion of times that the spline-based SCB captures the estimated CBIF from the Gompertz model.

5.2 Results

Figure 4 shows the plots of the RelRMSE as a function of time using the I-spline model and the Gompertz model to fit the data under the three scenarios. As we can see from the figure, the RelRMSE is generally decreasing as the sample size increases for the I-spline model across all of the three scenarios. When the sample size is large, the RelRMSE for the I-spline model estimator is within a small range.

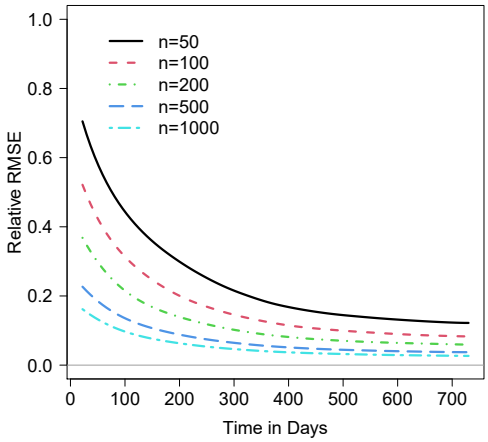
The behavior of the RelRMSE for the Gompertz model depends on the scenario. For Scenario 1, in which the Gompertz model fits well to the true model, the I-spline model tends to have a higher RelRMSE because there are more parameters in the I-spline model (and thus more variability in the estimates) than in the parametric model. For Scenario 2, in which there is some departure in the late stage, the Gompertz model has smaller RelRMSE than the I-spline model but the advantage diminishes when the sample size increases because bias begins to dominate variance. For Scenario 3, in which there is a large difference from the true model, the Gompertz model tends to have larger RelRMSE than the I-spline model and the RelRMSE does not decrease much when the sample size increases due to the effect of large bias.

In summary, the ML estimator for the I-spline model works as expected. When a parametric CBIF is adequate, it tends to have higher statistical efficiency. When the parametric model is not adequate, there could be large RelRMSE due to bias in the estimation. The results show that it is important to assess the adequacy of parametric models. The I-spline model, however, is flexible at the price of losing some statistical efficiency.

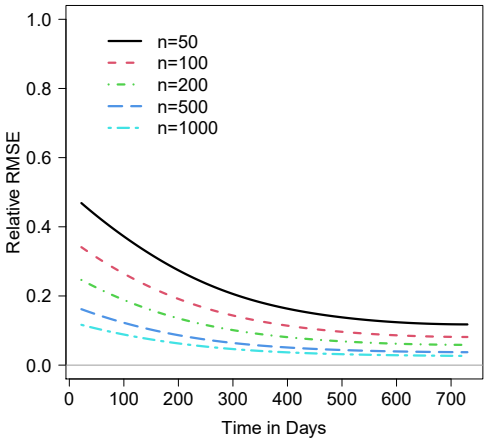
Figure 5 shows the plot of CP and acceptance probability as a function of sample size under the three scenarios. Figure 5(a) shows that the CP for the SCB procedure based on the I-spline model and bootstrap are similar for the three scenarios. The CP improves when the sample size increases. The CP is less than nominal when the sample size is small and is getting closer to the nominal CP when the sample is larger than 200. From the plot of the acceptance probability in Figure 5(b), the parametric model is generally accepted when the sample size is small and there is little or no departure from the true model. When there is a large departure from the true model, as in Scenario 3, the SCB does not capture the estimated Gompertz model with a high probability when the sample size is large.

6 Data Analysis

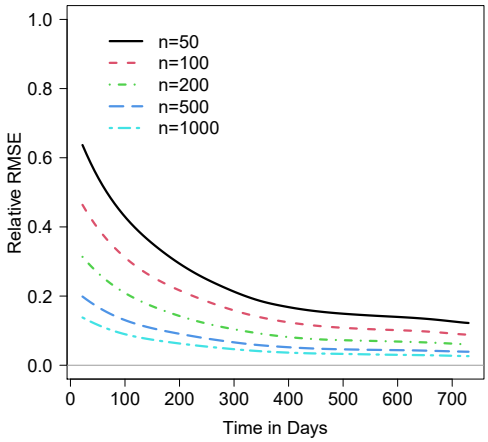
In this section, we present the data analysis for the recurrent disengagement events data.



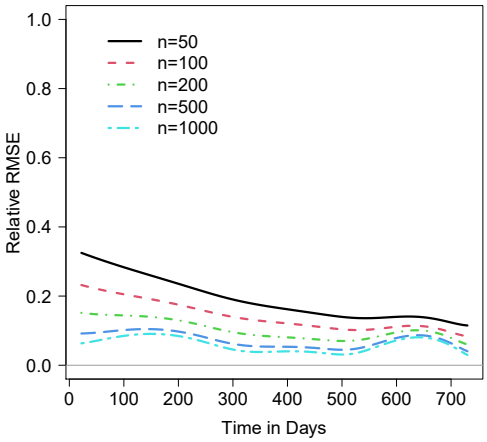
(a) Scenario 1, I-spline Model



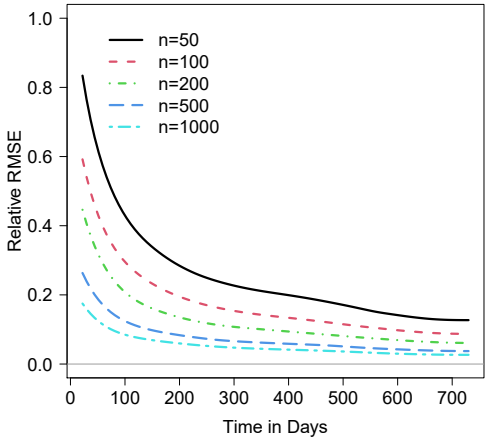
(b) Scenario 1, Gompertz



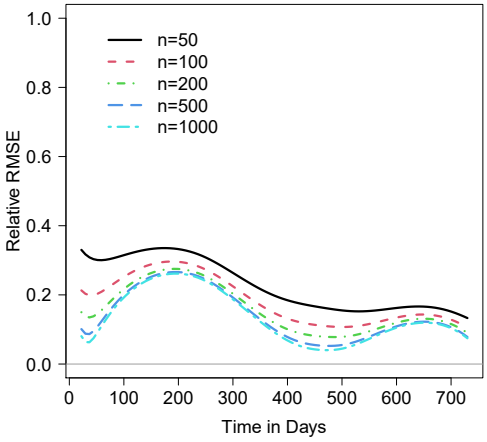
(c) Scenario 2, I-spline Model



(d) Scenario 2, Gompertz

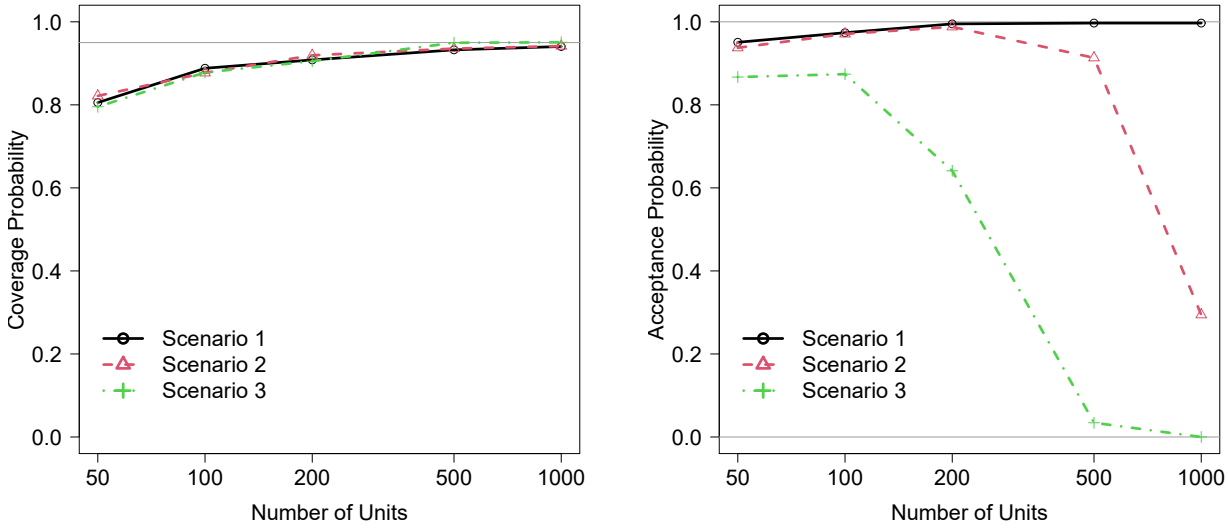


(e) Scenario 3, I-spline Model



(f) Scenario 3, Gompertz

Figure 4: Plots of relative RMSE as a function of time using the I-spline model and the Gompertz model to fit the data under the three scenarios.



(a) Coverage Probability (b) Acceptance Probability

Figure 5: Plots of coverage probability and acceptance probability as a function of sample size under the three scenarios.

6.1 Model Fitting

We start by fitting the parametric models in Table 2 and the I-spline model to the disengagement-events data from the four manufacturers: Waymo, Cruise, Pony AI, and Zoox. The model fitting uses the ML estimation procedures described in Section 4. Table 3 shows AIC values for the selected models fitted to the data from the four manufacturers. The numbers with bold font indicate the lowest AIC values among the parametric models. The I-spline model, in general, results in much lower AIC values except for Zoox data. This demonstrates that the I-spline model is flexible in fitting the recurrent event data. For Zoox, the AIC value for the Musa-Okumoto model is small because the model has two parameters. Based on the AIC values, the best parametric model for Waymo and Cruise is the Gompertz model. The best parametric model for Pony AI is the Weibull model, and the best parametric model for Zoox is the Musa-Okumoto model.

To visualize the model estimation results, Figure 6 shows the plots of the estimated CBIFs for the four manufacturers based on the I-spline model and parametric models, together with the 95% SCB based on the I-spline model. For Waymo and Cruise, all the parametric models are within the SCB and agree quite well with the estimated CBIF from the I-spline model. The Gompertz model agrees with the observed number of events better than other models over certain time ranges.

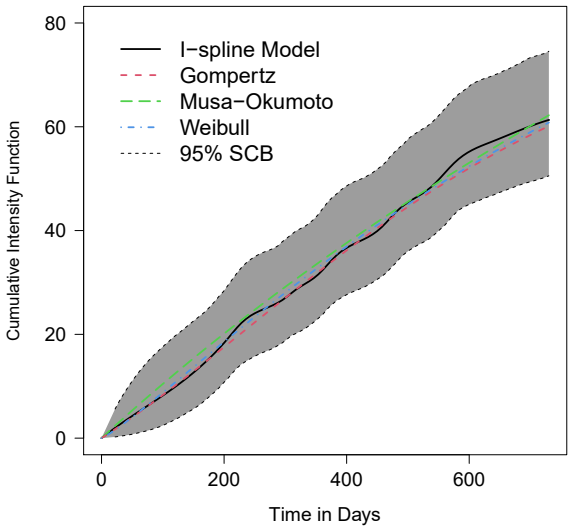
For Pony AI, the SCB is wide, indicating large variability in the estimation. In the early

Table 3: The values of AIC for fitting various models to the data from the four manufacturers. The numbers with bold font indicate the lowest AIC values among the parametric models.

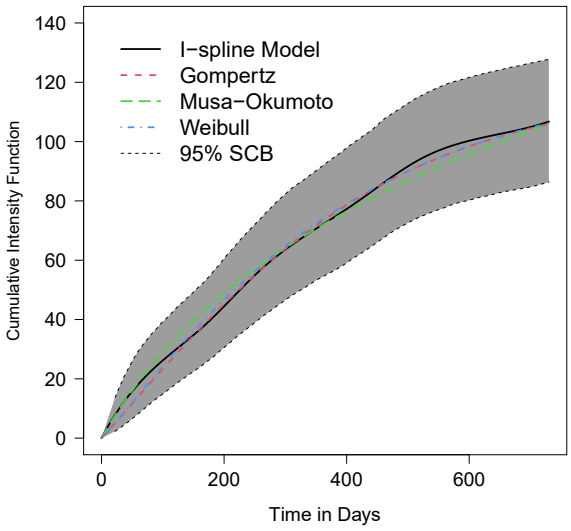
Manufacturer	Parametric Models			I-spline Model
	Musa-Okumoto	Gompertz	Weibull	
Waymo	2769.78	**2769.70**	2770.60	2756.21
Cruise	2051.27	**2047.42**	2048.28	2046.09
Pony AI	499.79	504.56	**498.73**	479.73
Zoox	**687.69**	689.55	689.38	688.78

stage of the testing (i.e., from day 0 to day 200), all of the events were coming from two units with a lower mileage driven at $x_i(t) = 0.01$. A high number of events with a low mileage driven leads to a high event rate. The SCB is asymmetric because it was built using the fractional-random-bootstrap and the distribution of $\widehat{\Lambda}_0(t)$ is heavily skewed. The fact that all events come from two units leads to a large amount of variability in the SCB as the change of weights of the two units has a large influence on the re-weighted log-likelihood. We also note that the best parametric model (the Weibull) does not agree well with the I-spline model, although it tracks the trend. For Zoox, the SCB is also relatively wide due to the small number of test units. All parametric models are within the SCB, indicating all parametric models are statistically acceptable.

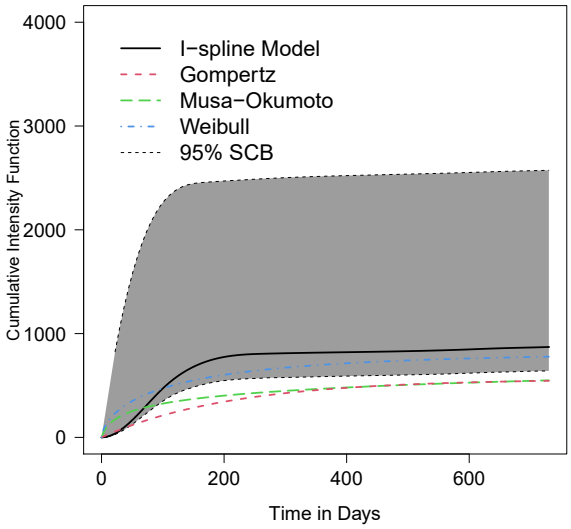
To further check how well the models fit the data, Figure 7 shows the plots of the expected versus the observed number of events for the four manufacturers based on the I-spline model and parametric models, together with the 95% PCIs based on the I-spline model. The expected number of events is computed based on the specific model with an adjustment for the mileage history from all units. The PCIs for the expected number of events are based on bootstrap samples. The shape of the function of the cumulative number of events differs from the shape of the CBIF because the function of the cumulative number of events is adjusted by the rate $x_i(t)$, which is time-varying and depends on the driving pattern, while CBIF is the case when $x_i(t) = 1$. In all cases, the I-spline model tracks the cumulative number of observed events well. For Waymo, Cruise, and Zoox, the Gompertz and Weibull models also track the counts well, but visually we can see some departures for the Musa-Okumoto model. For Pony AI, all three parametric models show significant departures in the plot, indicating they are not flexible enough to describe that event process.



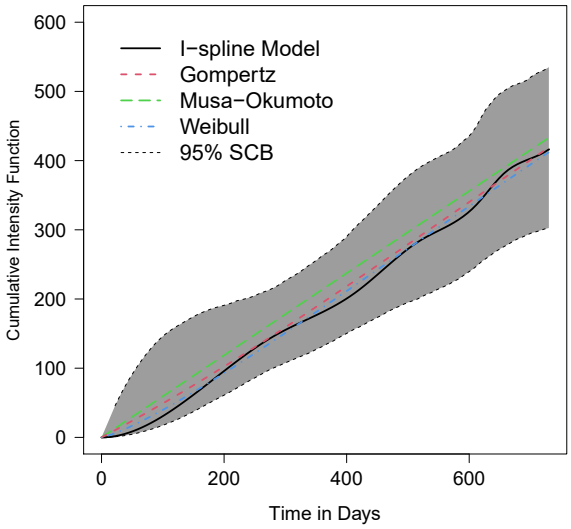
(a) Waymo



(b) Cruise

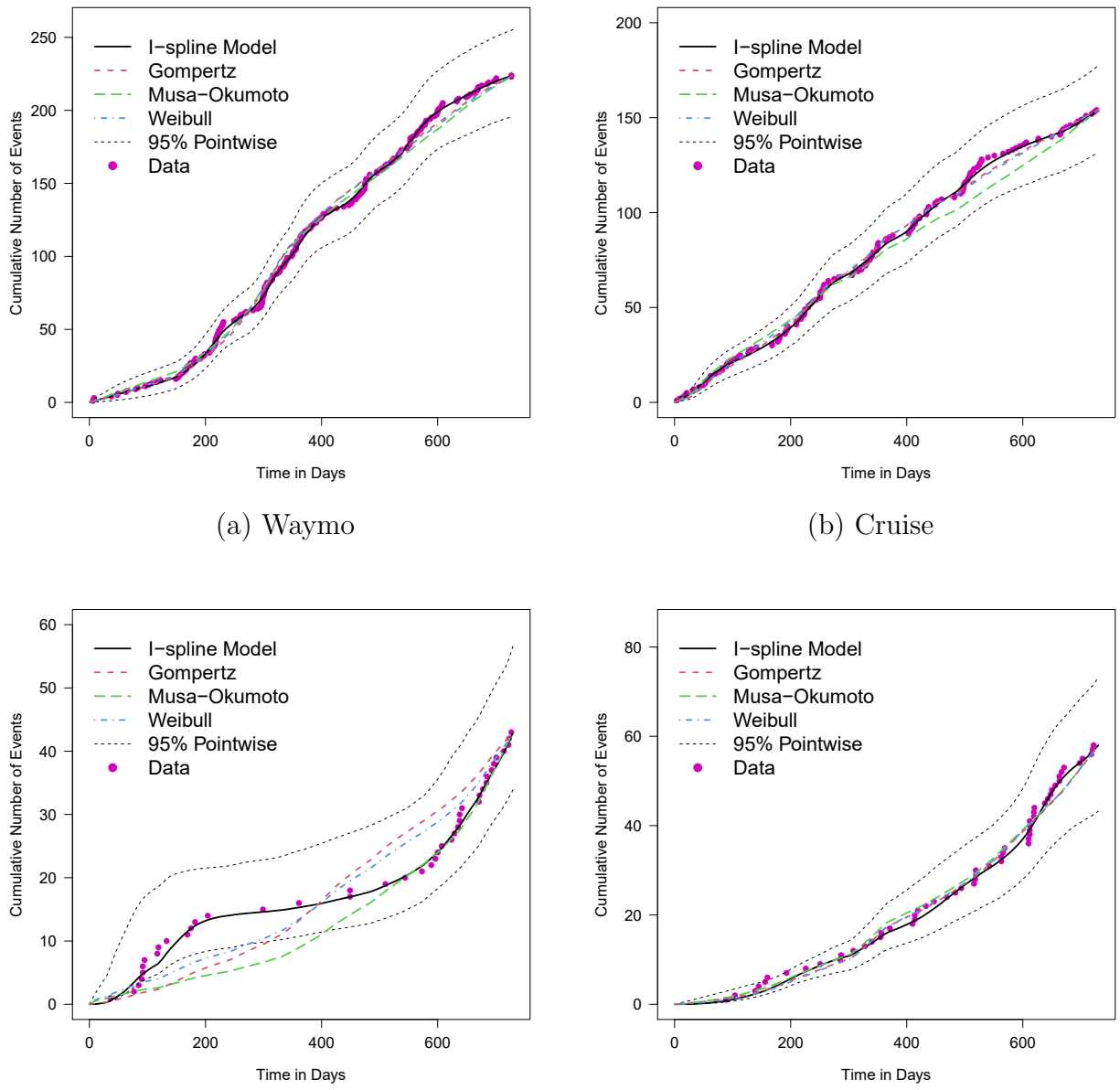


(c) Pony AI



(d) Zoox

Figure 6: Plots of the estimated CBIFs for the four manufacturers based on the I-spline model and parametric models, together with the 95% SCB based on the I-spline model.



(a) Waymo (b) Cruise

(c) Pony AI (d) Zoox

Figure 7: Plots of the expected versus the observed number of events for the four manufacturers based on the spline and parametric models, together with the 95% PCIs based on the I-spline model.

6.2 Results Interpretation

Table 4 shows the parameter estimates, standard errors, and approximate 95% confidence intervals (CIs) for the best parametric models for the four manufacturers. The CIs for parameters are based on a normal (Wald) approximation, and some transformations (e.g., a logarithm transformation for positive parameters) are used to improve the performance. Note that the estimate for θ_1 for the Zoox/Musa-Okumoto is set to zero because the model is degenerate at $\theta_1 = 0$, indicating a constant rate situation.

We also tested population heterogeneity using the procedure in Section 4.3. Table 5 shows the summary of the gamma frailty models for the four manufacturers. All of the p-values are close to 1, indicating little population heterogeneity among the event processes for the four manufacturers. Hence, it is reasonable to use a model for which all units have the same event process with the same CBIF. Table 5 also lists the names of the best parametric models for each manufacturer.

To further visualize the results, Figure 8 plots the BIFs from the best parametric models for each of the four manufacturers. From the plot, we see the trends for the different manufacturers. A decreasing trend means an improvement in AI technology and an increase in AV reliability. Waymo, Cruise, and Pony AI display a decreasing trend, while Zoox displays a constant rate of about 0.6 events per k-miles. Waymo starts at an event rate near 0.1 events per k-miles and decreases over the two-year period. Cruise starts at 0.2 events per k-miles and shows a decreasing trend. Pony AI starts at a high rate of 10 events per k-miles and shows a rapidly improving rate. By the end of the study period (i.e., November 30, 2019), the event rate for Waymo and Cruise is around 0.05 events per k-miles. The event rate for Pony AI is around 0.1 events per k-miles. This pattern indicates that there is a lot of improvement for the reliability of the Pony AI driving system. From the analysis conducted in this section for different manufacturers, we can see that the overall AV reliability is improving over the two-year period.

As a comparison, Figure 9 shows the estimated BIFs based on the I-spline model and the parametric models, and the 95% PCIs based on the I-spline model. We can see that the I-spline model shows more variation but the general trends are the same as the parametric models. The estimates for Zoox have a large amount of variability due to the limited sample size, as discussed previously. Also, the BIFs are estimated to be 0 when $t = 0$ for Pony AI and Zoox and non-zero for Waymo and Cruise.

6.3 Comparisons Among Different Methods

Besides our proposed I-spline method, there are other nonparametric methods that can also be used to estimate the BIF or CBIF. Here we compare our proposed method with three

Table 4: Parameter estimates, standard errors, and approximate 95% CIs for the best parametric models for the four manufacturers.

Manufacturer/ Model	Parameter	Estimate	Std. Err.	95% CI	
				Lower	Upper
Waymo	θ_1	102.2539	31.7600	55.6278	187.9610
/	θ_2	0.9975	0.0009	0.9951	0.9987
Gompertz	θ_3	0.1623	0.1229	0.0319	0.5326
Cruise	θ_1	171.3352	66.1936	80.3503	365.3472
/	θ_2	0.9963	0.0009	0.9941	0.9977
Gompertz	θ_3	0.3064	0.2285	0.0510	0.7842
Pony AI	θ_1	817.203	273.828	423.744	1575.997
/	θ_2	0.0474	0.0474	0.0067	0.3363
Weibull	θ_3	0.6304	0.1615	0.3815	1.0416
Zoox/	θ_1	0.0000	0.0000	0.0000	0.0000
Musa-Okumoto	θ_2	0.5933	0.0779	0.4587	0.7674

Table 5: Summary of the gamma frailty models for the four manufacturers.

Manufacturer	Best Parametric Model	Variance $(\widehat{\phi})$	p-value
Waymo	Gompertz	0.0001	0.9721
Cruise	Gompertz	0.0000	0.9972
Pony AI	Weibull	0.0000	0.7229
Zoox	Musa-Okumoto	0.0197	0.8758

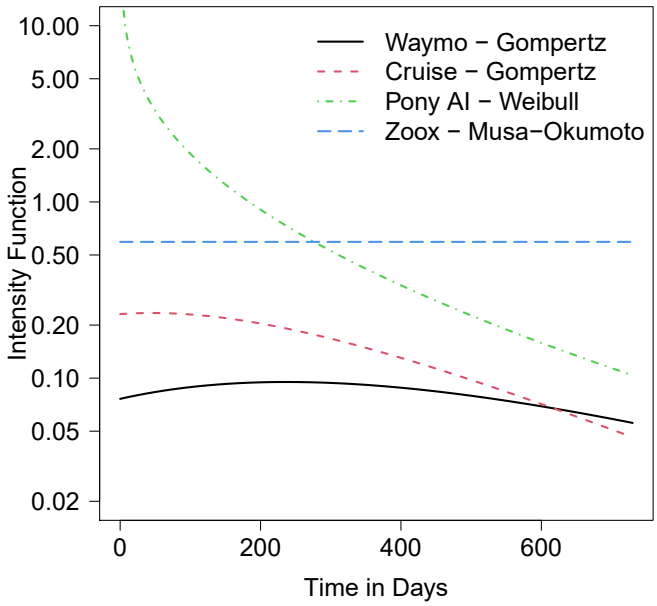
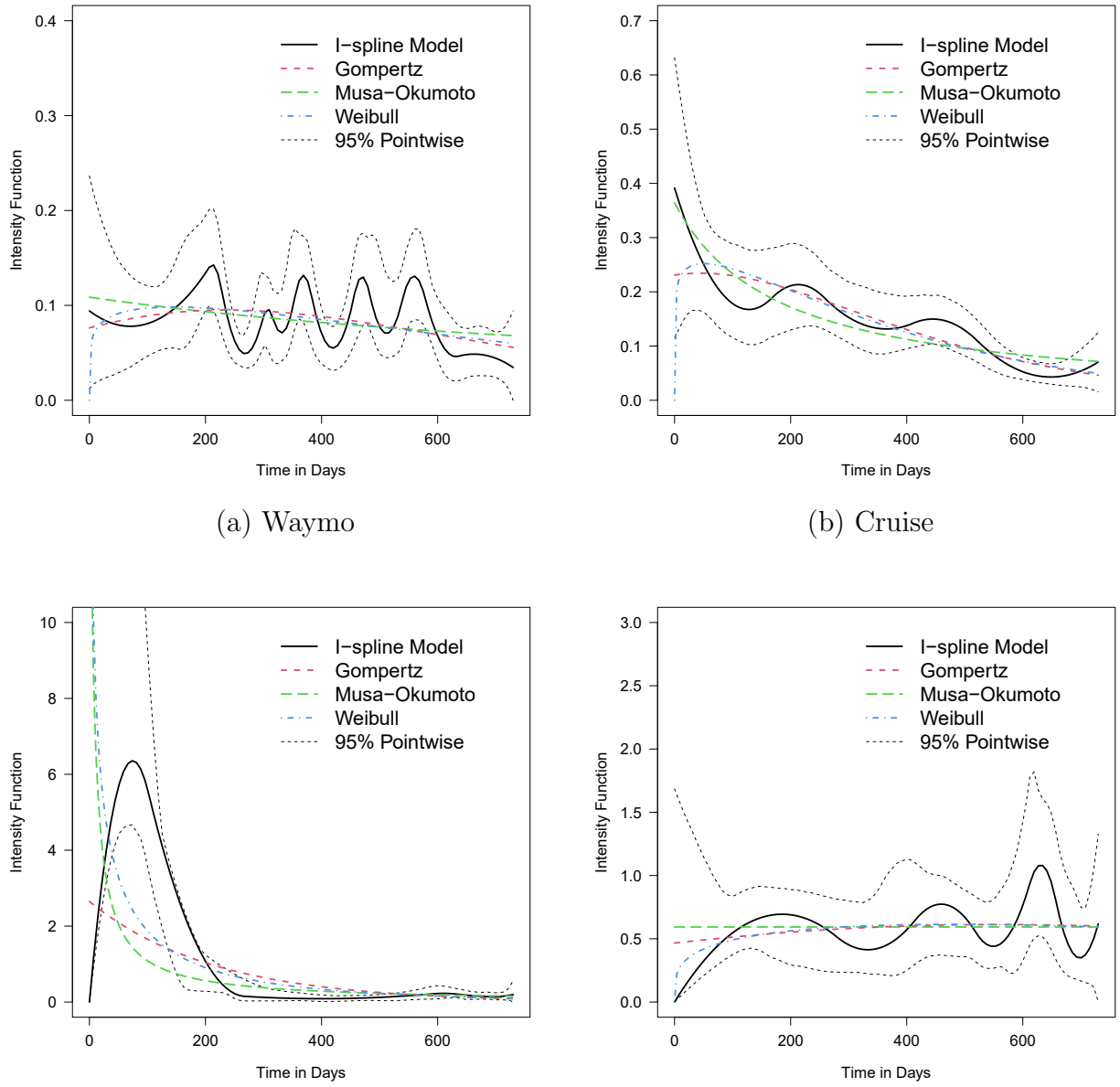


Figure 8: Plot of BIFs from the best parametric models for the four manufacturers.

other different nonparametric methods: the logBIF method which directly models the log of the BIF, the P-spline method, and the piecewise constant method.

Morgan et al. (2019) stated that it is common to model the logarithm of BIF because of the non-negativity of BIF. Because we use I-splines to model the CBIF in this paper, it is intuitive to use M-splines to model the logarithm of BIF, so that there is no need to constrain the spline coefficients for the positiveness of the BIF. To be consistent with our proposed I-spline method, we use M-splines of order 3, and the interior knots are again set to be equally spaced sample quantiles of the observed event times. We use AIC to select the number of knots, similar to the I-spline method.

The P-spline method uses penalized B-spline with constrained coefficients to model the CBIF. Marra and Radice (2019) proposed to use P-splines with monotonic constraints to model the baseline survival function of time-to-event data with R implementation available in package GJRM (Marra and Radice 2021). Using a similar idea, we combined the P-splines with monotonic constraints and penalized likelihood to estimate the CBIF for our setting. Because $\Lambda_0(0;\boldsymbol{\theta}) = 0$ and $\Lambda_0(t;\boldsymbol{\theta}) > 0$ when $t > 0$, we exclude the intercept in B-spline basis, and put a constraint on the first spline coefficient to be greater than 0. Interior knots are set to be equally spaced. In estimation, we use AIC to select both the number of knots and the value of the penalty term. In calculating AIC, the effective degrees of freedom is calculated as described in Therneau and Grambsch (2000, page 121). Because we use a penalized spline



(a) Waymo

(b) Cruise

(c) Pony AI

(d) Zoox

Figure 9: Plots of the estimated BIFs for the four manufacturers based on the I-spline model and parametric models, together with the 95% PCIs based on the I-spline model. The "wiggliness" of the estimated BIFs in (a) mainly comes from data, as suggested by the estimation results from the piecewise model as shown in Figure 10.

and equal spacing to place the interior knots, we consider different sets of the number of knots compared with the I-spline method and the logBIF method. The number of interior knots for Waymo and Cruise is from 1 to 36, the number of interior knots for Pony AI is from 1 to 19, and the number of interior knots for Zoox is from 1 to 13. The value of penalty terms to be considered is 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 15, 20, 50 and 100 for all the four manufactures.

The piecewise constant method is another popular method to model the BIF. It fixes the BIF as constants over different time intervals. Unlike Chen and Schmeiser (2013), who smoothed the estimation of piecewise constant, we fit a constant BIF within each month. We selected the one-month interval because the monthly mileage of each testing vehicle is made available to the public, and it is natural to consider monthly interval when the data are reported monthly. Because the choice of length of intervals influences the wiggliness of the estimated BIF, we also tried two-month and three-month intervals, which are shown in Supplementary Figures 1 to 4. We did not go beyond three months because that would result in too few time points for estimation. Using the three-month interval provides a smoother estimated BIF, but one can still see some wiggliness in data from certain manufacturers (e.g., Zoox). The fitted piecewise constant model can be viewed as a type of nonparametric estimate that follows the data.

Figure 10 shows the comparison among the estimated BIFs using the proposed method, the logBIF method, the P-spline method, and the piecewise constant method. The advantage of the logBIF method is there is no constraint on the coefficients, and confidence intervals can be calculated using a normal approximation. The disadvantage is the computing time of optimization can be long for the logBIF method, because the integral of BIF to obtain CBIF needs to be calculated at every iteration in optimization. Using the proposed method to estimate CBIF, the integration only needs to be calculated once for computing the I-spline bases. The estimation results using the logBIF method and our proposed method are comparable for Cruise, Pony AI, and Zoox. For the Waymo, the logBIF tends to have more "wiggle" in its estimation. This is partly caused by the fact that the exponential function magnifies the effect of spline coefficients.

Figure 10 shows that both the I-spline and the P-spline models can provide a good fit to the data, and the proposed I-spline model provides a better track of the trend in some cases. In addition, the result from the piecewise constant model indicates that the "wiggliness" of the estimated BIF from those spline models mainly comes from the data.

We also investigated the computing time for parameter estimation using various methods, which is shown in Supplementary Table 1. The piecewise constant model is the fastest one due to the closed-form expression for the parameter estimator. The proposed I-spline method is after the piecewise constant method. The P-spline method takes a little bit longer due to

selecting both the number of knots and the value of the penalization parameter using AIC. We also note that the logBIF method is slow due to the need to integrate the BIF to obtain the CBIF in each iteration.

7 Conclusions and Areas for Future Research

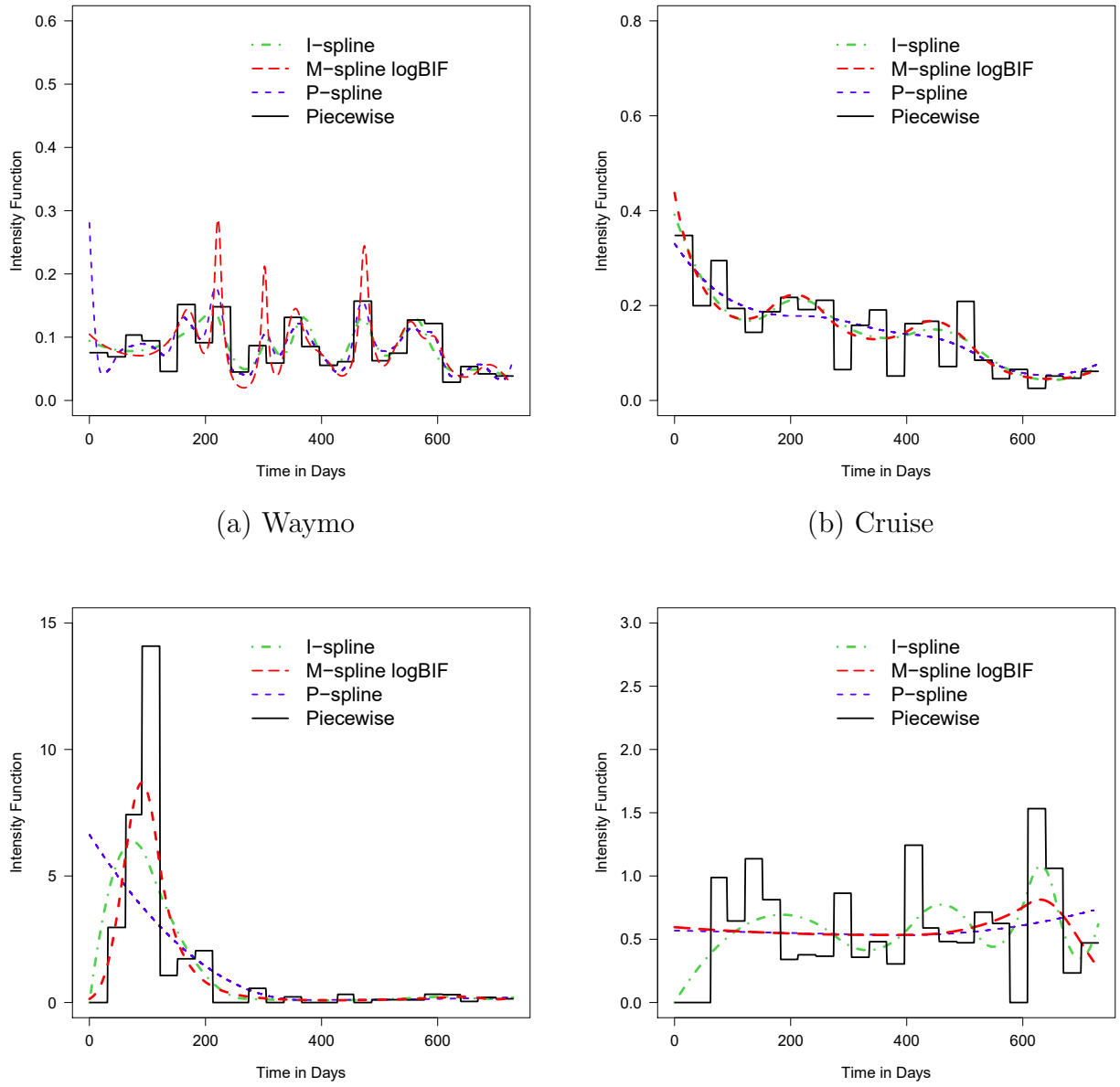
This paper focuses on the reliability analysis of AV technology using the recurrent disengagement events from the California driving study. We propose a statistical framework for modeling and analyzing the recurrent events data from AV driving tests. We use both parametric models and a nonparametric model to describe the event processes. Based on the I-spline model, we can select the best model, quantify uncertainty, and test heterogeneity in the event process. We want to point out that the parametric models and spline models are proposed as complementary tools for modeling and inference.

The simulation and data analysis show that the proposed spline model is flexible for describing the recurrent events data from four AV manufacturers, and the parametric models are adequate for data from most manufacturers. It is worth noting that the best parametric models can be different for different manufacturers. The population heterogeneity in the event process is also low. From the data analysis, we found that the overall AV reliability is improving over the two-year study period.

The currently available data do not include covariates, such as the driving speed when the event occurred, the test environment (e.g., busy street versus freeway), and vehicle models. In the future, it would be interesting and useful to collect more covariate information. Our proposed modeling framework can be extended to analyze recurrent disengagement events data with covariates. The CA DMV recently started a driverless program where cars on the road do not require a driver. It would be interesting to analyze the driverless study data in the future when enough event data are available.

Another possible future research topic is the combination of the frailty term and the spline model. Frailty models are typically more difficult to estimate because frailties are not observed, and spline methods are flexible to capture time trends. The combination of the two looks appealing, but practically, it is challenging to implement. Under the spline method, the CBIF is modeled as a linear combination of spline bases. Multiplying a frailty term to the CBIF can lead to difficulty in estimation and parameter identifiability problems that might need special treatment. It is an interesting question to study further.

In addition, from the comparison of different methods, it is worthy to note that the P-spline method can be a good choice when the data itself is wiggly. In this scenario, the P-splines can smooth the fluctuation from the data and provide a smoother estimation compared with



(a) Waymo

(b) Cruise

(c) Pony AI

(d) Zoox

Figure 10: Plots of the estimated BIFs based on the I-spline, logBIF, P-spline, and piecewise constant models.

the proposed I-spline method. For example, Figure 10(d) shows this advantage of P-splines, in which the estimated BIF using I-splines fluctuates with the data, while the estimated BIF using the P-spline method is much smoother. Also, it is interesting to look into other methods, such as those with automatic smoothing parameter estimation and fixed knots (e.g., Wood, Pya, and Säfken 2016). Instead of using AIC, automatic smoothing methods could lead to smoother estimation of the CBIF and is worthy of being investigated in future research.

Because reliability is a property that evolves over time, all kinds of AI systems need to be tested over time to quantify their reliability. Although our analysis focuses on AV reliability, our proposed data analytic framework can also be applied to assess the reliability of other AI systems where departures from desired behavior provide recurrent events data. With an appropriate definition of time scale and events, the parametric and spline models discussed in this paper can be extended to analyze data from the reliability testing of other AI systems.

Computing hardware reliability is also an important aspect of AI reliability. For example, GPUs are widely used in AI model computing. Ostrouchov et al. (2020) considered the lifetimes of GPUs used supercomputers. It would be interesting to study GPU reliability, or more broadly computing hardware reliability, and its relationship to AI reliability.

Supplementary Material

The following supplementary materials are available online.

Additional details: Additional results for data analysis (pdf file).

Code and data: R code for data analysis and simulations. The disengagement events data used for analysis are also included (zip file).

Acknowledgments

The authors thank the editor, associate editor, and two referees, for their valuable comments that helped improve the paper significantly. The authors acknowledge the Advanced Research Computing program at Virginia Tech for providing computational resources. The work by Hong was partially supported by National Science Foundation Grant CMMI-1904165 to Virginia Tech.

References

Alshemali, B. and J. Kalita (2020). Improving the reliability of deep neural networks in NLP: A review. *Knowledge-Based Systems 191*, 105210.

Amodei, D., C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mane (2016). Concrete problems in AI safety. *arXiv: 1606.06565*.

Åsljung, D., J. Nilsson, and J. Fredriksson (2017). Using extreme value theory for vehicle level safety validation and implications for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles 2*, 288–297.

Banerjee, S. S., S. Jha, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer (2018). Hands off the wheel in autonomous vehicles?: A systems perspective on over a million miles of field data. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 586–597. IEEE.

Boggs, A. M., B. Wali, and A. J. Khattak (2020). Exploratory analysis of automated vehicle crashes in California: A text analytics & hierarchical Bayesian heterogeneity-based approach. *Accident Analysis and Prevention 135*, 105354.

Bosio, A., P. Bernardi, A. Ruospo, and E. Sanchez (2019). A reliability analysis of a deep neural network. In *2019 IEEE Latin American Test Symposium (LATS)*, pp. 1–6.

Burke, K., M. C. Jones, and A. Noufaily (2020). A flexible parametric modelling framework for survival analysis. *Journal of the Royal Statistical Society: Series C 69*, 429–457.

Burton, S., I. Habli, T. Lawton, J. McDermid, P. Morgan, and Z. Porter (2020). Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective. *Artificial Intelligence 279*, 103201.

California Department of Motor Vehicles. Autonomous vehicle tester program. [Online]. Available: `https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/`, accessed: September 01, 2020.

Chen, H. and B. Schmeiser (2013). I-smooth: Iteratively smoothing mean-constrained and nonnegative piecewise-constant functions. *INFORMS Journal on Computing 25*(3), 432–445.

Cook, R. J. and J. F. Lawless (2007). *The Statistical Analysis of Recurrent Events*. New York: Springer-Verlag.

Cruise. [Online]. Available: `https://www.getcruise.com/`, accessed: September 01, 2020.

Dixit, V. V., S. Chand, and D. J. Nair (2016). Autonomous vehicles: disengagements, accidents and reaction times. *PLoS One 11*, e0168054.

Duchateau, L. and P. Janssen (2008). *The Frailty Model*. New York: Springer-Verlag.

Ehrlich, W. K., V. N. Nair, M. S. Alam, W. H. Chen, and M. Engel (1998). Software reliability assessment using accelerated testing methods. *Journal of the Royal Statistical Society: Series C 47*, 15–30.

Favarò, F., S. Eurich, and N. Nader (2018). Autonomous vehicles disengagements: Trends, triggers, and regulatory limitations. *Accident Analysis & Prevention 110*, 136–148.

Goldstein, B. F., S. Srinivasan, D. Das, K. Banerjee, L. Santiago, V. C. Ferreira, A. S. Nery, S. Kundu, and F. M. G. França (2020). Reliability evaluation of compressed deep learning models. In *2020 IEEE 11th Latin American Symposium on Circuits Systems (LASCAS)*, pp. 1–5.

Hong, Y., L. A. Escobar, and W. Q. Meeker (2010). Coverage probabilities of simultaneous confidence bands and regions for log-location-scale distributions. *Statistic & Probability Letters 80*, 733–738.

Hong, Y., M. Li, and B. Osborn (2015). System unavailability analysis based on window-observed recurrent event data. *Applied Stochastic Models in Business and Industry 31*, 122–136.

Huang, C.-Y., M. R. Lyu, and S.-Y. Kuo (2003). A unified scheme of some nonhomogenous Poisson process models for software reliability estimation. *IEEE Transactions on Software Engineering 29*, 261–269.

Jin, Z., Z. Ying, and L. J. Wei (2001). A simple resampling method by perturbing the minimand. *Biometrika 88*, 381–390.

Kalra, N. and S. M. Paddock (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice 94*, 182–193.

Lawless, J. F. (1995). The analysis of recurrent events for multiple subjects. *Journal of the Royal Statistical Society. Series C 44*, 487–498.

Lawless, J. F. (2003). *Statistical Models and Methods for Lifetime Data* (2nd ed.). New Jersey, Hoboken: John Wiley & Sons, Inc.

Lv, C., D. Cao, Y. Zhao, D. J. Auger, M. Sullman, H. Wang, L. M. Dutka, L. Skrypchuk, and A. Mouzakitis (2018). Analysis of autopilot disengagements occurring during autonomous vehicle testing. *IEEE/CAA Journal of Automatica Sinica 5*, 58–68.

Marra, G. and R. Radice (2019). Copula link-based additive models for right-censored event time data. *Journal of the American Statistical Association 115*, 886–895.

Marra, G. and R. Radice (2021). *Generalised Joint Regression Modelling*. R package version 0.2-5.1.

Meeker, W. Q., L. A. Escobar, and F. G. Pascual (2021). *Statistical Methods for Reliability Data* (Second ed.). New Jersey, Hoboken: Wiley.

Merkel, R. (2018). Software reliability growth models predict autonomous vehicle disengagement events. *arXiv: 1812.08901*.

Meyer, M. C. (2008). Inference using shape-restricted regression splines. *The Annals of Applied Statistics 2*, 1013–1033.

Michelmore, R., M. Wicker, L. Laurenti, L. Cardelli, Y. Gal, and M. Kwiatkowska (2019). Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. *arXiv: 1909.09884*.

Morgan, L. E., B. L. Nelson, A. C. Titman, and D. J. Worthington (2019). A spline-based method for modelling and generating a nonhomogeneous Poisson process. In *2019 Winter Simulation Conference (WSC)*, pp. 356–367. IEEE.

Musa, J. D. and K. Okumoto (1984). A logarithmic Poisson execution time model for software reliability measurement. In *Proceedings of the 7th International Conference on Software Engineering*, pp. 230–238. IEEE Press.

Nair, V. N. (1984). Confidence bands for survival functions with censored data: a comparative study. *Technometrics 26*, 265–275.

Ostrouchov, G., D. Maxwell, R. A. Ashraf, C. Engelmann, M. Shankar, and J. H. Rogers (2020). GPU lifetimes on Titan supercomputer: Survival analysis and reliability. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '20)*, New York, NY. Association for Computing Machinery.

Pony AI. [Online]. Available: `https://www.pony.ai/en/index.html`, accessed: September 01, 2020.

Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Science 3*, 425–441.

Rubin, D. B. (1981). The Bayesian bootstrap. *Annals of Statistics 9*, 130–134.

Shan, Q., Y. Hong, and W. Q. Meeker (2020). Seasonal warranty prediction based on recurrent event data. *Annals of Applied Statistics 14*, 929–955.

Therneau, T. M. and P. M. Grambsch (2000). *Modeling Survival Data: Extending the Cox Model*. New York: Springer.

Waymo. [Online]. Available: `https://waymo.com/`, accessed: September 01, 2020.

Wood, A. (1996). Software reliability growth models. Tandem Technical Report 96.1, Tandem Computers, Cupertino, CA.

Wood, S. N., N. Pya, and B. Säfken (2016). Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association 111*, 1548–1563.

Xie, Y., C. B. King, Y. Hong, and Q. Yang (2018). Semi-parametric models for accelerated destructive degradation test data analysis. *Technometrics 60*, 222–234.

Xu, L., C. Gotwalt, Y. Hong, C. B. King, and W. Q. Meeker (2020). Applications of the fractional-random-weight bootstrap. *The American Statistician 74*, 345–358.

Zhao, X., A. Banks, J. Sharp, V. Robu, D. Flynn, M. Fisher, and X. Huang (2020). A safety framework for critical systems utilising deep neural networks. *arXiv: 2003.05311*.

Zhao, X., V. Robu, D. Flynn, K. Salako, and L. Strigini (2019). Assessing the safety and reliability of autonomous vehicles from road testing. In *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 13–23.

Zoox. [Online]. Available: `https://zoox.com/`, accessed: September 01, 2020.

Zuo, J., W. Q. Meeker, and H. Wu (2008). Analysis of window-observation recurrence data. *Technometrics 50*, 128–143.