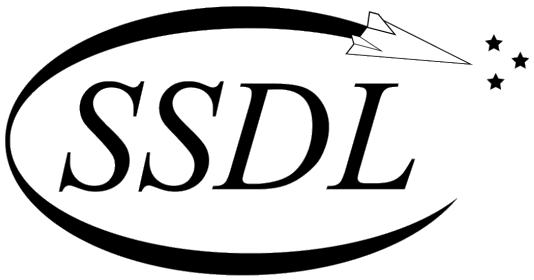

Space Systems Design Laboratory

A Statistical Analysis and Predictive Modeling of Safing Events for Interplanetary Spacecraft

Author:
Swapnil R. Pujari

Advisor:
Dr. E. Glenn Lightsey

AE 8900 MS Special Problems Report
Space Systems Design Laboratory (SSDL)
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, Georgia, 30332

April 27, 2018

This report is presented to the Academic Faculty by Swapnil R. Pujari under the advisement of Dr. E. Glenn Lightsey, in partial fulfillment of the requirements for the Degree Master of Sciences in Aerospace Engineering at the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology.

A Statistical Analysis and Predictive Modeling of Safing Events for Interplanetary Spacecraft

Swapnil Pujari
Georgia Institute of Technology
270 Ferst Drive
Atlanta, GA 30332
678-654-2569
spujari21@gatech.com

Glenn Lightsey
Georgia Institute of Technology
270 Ferst Drive
Atlanta, GA 30332
404-385-4146
glenn.lightsey@gatech.edu

Abstract— **Unexpected spacecraft failures and anomalies may prompt autonomous on-board systems to change a spacecraft's state to a 'safe mode' in order to isolate and resolve the problem. Future interplanetary missions such as Psyche and the proposed Next Mars Orbiter mission concept, plan to use solar electric propulsion on-board. Continuous operation of the thrusters is necessary in order to achieve their mission objectives. The motivation for this paper stems from a need to better predict safing events based on various mission factors such as mission class, destination, duration, etc. Modeling spacecraft inoperability due to a spacecraft entering safe mode is imperative in order to appropriately allocate spacecraft margins and shape design & operations requirements. This paper contributes to the area of safing events by further analyzing trends and dependencies within the available data subsets, and develops predictive models of frequency and recovery times of safing events for interplanetary spacecraft missions.**

First, the full safing event dataset is split into multiple subsets based on various mission classifiers. By employing the Chi-Squared hypothesis test, the degree of dependency between classifiers is assessed. A parametric analysis is conducted using a single and mixture of two Weibull distributions. The optimal parameters that would best fit the full dataset and subsets are computed by a maximization likelihood algorithm. The mean square error and Akaike Information Criteria represent goodness-of-fit criteria for the computed distributions; insight into any inherent bi-modal behavior is identified through these criteria. A supervised learning algorithm is utilized in capturing and understanding relationships between input and output variables, and utilizing these to predict unknown outcomes. For the safing event database, two Gaussian process models are trained, tested, and deployed: one for time-between-events and the other for recovery durations. By incorporating these Gaussian Process models into a mission simulation framework, a Monte Carlo simulation of the likelihood of inoperability rates is conducted to robustly predict safing events. A greater understanding of the safing event dataset through statistical & parametric analyses, and the development of a Gaussian Process model for predictions enables interplanetary mission planners to make more informed decisions during spacecraft development.

TABLE OF CONTENTS

978-1-5386-2014-4/18/$31.00

1. INTRODUCTION & MOTIVATION

Advances in on-board computing in the last few decades have enabled robotic spacecraft missions to take control of tasking and autonomous responsibilities with less interactions from the ground. Although engineers thoroughly design and test a variety of conditions faced by the spacecraft, unexpected failures and anomalies may still arise during the mission lifetime. Rather than letting the spacecraft operate in such a state, a 'safe' mode can be implemented which is when the spacecraft's systems are preserved until the ground can diagnose and recover from the situation. Safe mode is typically defined as the state in which non-essential components and subsystems are powered off, while the spacecraft maintains an attitude such that it is power positive, thermally stable, and commandable by ground operators [1]. As more complex missions are developed, the need for greater on-board autonomy increases.

The proposed Next Mars Orbiter (NeMO) mission concept is one such example and a case study for this paper. NeMO may support relay & telecommunications in the Martian relay network, perform remote sensing of Mars, and partake in the Mars Sample Return campaign [2]. It may include high-power, high-Isp solar electric propulsion (SEP) to increase the overall capability of the mission. The advantage of SEP is that a relatively small propellant mass is needed for ΔV maneuvers compared to chemical propulsion systems. NeMO is not the first SEP interplanetary mission; JPL has flown SEP on Deep Space 1 (DS1) and Dawn. Furthermore, JPL has baselined SEP technology on the Pysche mission that is planned to launch in 2022.

During NeMO's interplanetary transfer to Mars, the SEP engines will need to operate continuously to achieve the necessary ΔV since they produce significantly lower thrust than chemical engines. Multiple thrusting segments lasting weeks to months may be necessary during the interplanetary cruise phase of the mission. This requires the spacecraft to remain fully operational during this extended maneuver, placing requirements on the spacecraft to operate in an autonomous manner such that it does not interrupt these thrusting arcs. If the spacecraft enters safe mode, those safing events have the effect of reducing overall operability. The frequency and recovery time of safing events may lengthen the mission and increase risk in the ability to fulfill its full mission success criteria. A characteristic typical to all SEP, inoperability is a metric that significantly shapes the design and margins of a spacecraft. Typical inoperability values have been estimated using best engineering practices; however, developing a more

rigorous analysis and predictive methodology allows to accurately quantify the likelihood and effects of safing events on spacecraft operability.

An interplanetary spacecraft safe mode analysis was first done by Imken et al. [3]. A database of 240 safe mode entries from 21 interplanetary spacecraft was collected through a variety of sources including the Jet Propulsion Laboratory (JPL), NASA's Goddard Space Flight Center (GSFC), NASA's Ames Research Center (Ames), and Johns Hopkins University Applied Physics Laboratory (JHUAPL). This database contains missions starting with the Galileo mission, launched in 1989, and continues to present day with active missions. It not only includes when the safing event occurred but also mission statistics, root cause of the event, event recovery timeline, and other relevant data. The definitions of time between events, recovery duration, and inoperability period developed by Imken et al. are used in this paper in the same manner. Imken et al. also developed a Monte Carlo simulation to predict the likelihood of realizing an inoperability rate for future missions using the interplanetary safing event dataset [3]. A majority of the simulation framework is used in this paper, but is modified to include the developed model to generate alternate frequency and recovery duration predictions.

The modeling and distribution fitting work done by Imken et al. indicates that the Weibull distribution is a good candidate for the time-between-events and recovery duration datasets [3]. Due to its flexibility in describing a dataset with just two parameters, a Weibull distribution is commonly used in reliability models. Castet and Saleh modeled satellite reliability for approximately 1600 Earth-orbiting satellites using both nonparametric and parametric models [4]. A Weibull parametric model was shown to best fit the nonparametric satellite failure data. Mixed Weibull distributions, a linear combination of two Weibull distributions, can also provide modeling nonparametric satellite reliability with greater accuracy, as was done by Dubos et al. [5].

Predictive analytics is an area of statistics that deals with obtaining data about a system and using it to predict future trends for a particular application. Predictive analytics can be defined as, "Technology that learns from experience (data) to predict the future behavior of individuals in order to drive better decisions" [6]. Although no formal process exists, there are several general steps when applying predicting analytics for a certain dataset [7].

(1) **Project Definition**: Define the objectives, outcomes, scope of effort, and data sets needed to translate into tasks;

(2) **Data Collection**: Mine and collect all relevant data from as many sources as needed;

(3) **Data Preparation**: Inspect, simplify, and clean the data upon which to analyze and create models;

(4) **Statistical Analysis**: Using standard statistical methodologies, validate the assumptions using hypothesis tests to understand the dataset;

(5) **Predictive Modeling**: Choose, create, implement, test, and validate a model to generate results for prediction purposes;

(6) **Model Deployment**: Apply the model to the the particular outcome or case study; and

(7) **Model Monitoring**: Manage the model and repeat any steps necessary to improve performance.

In the scheme of predictive analytics for interplanetary safing events, Imken et al. has successfully completed the first three steps by defining the scope, collecting and preparing the dataset, and statistically determining that a Weibull distribution models the time-between-events and recovery duration. The foundation for the work done in this paper starts with the dataset collection, data modeling, and simulation efforts done by Imken et al. [3].

This paper contributes to the study of interplanetary spacecraft safing events by exploring the statistical properties of the dataset and developing predictive models as defined by steps 4, 5, and 6 in the context of predictive analytics. First, the dataset and subsets from all mission inputs is created, and a statistical independence test is performed between each input. A parametric analysis is conducted by creating single and mixed Weibull distributions and evaluating the goodness-of-fit using multiple criteria. Then, a generalized predictive model using Gaussian Process models with varying mission inputs is trained and tested by selecting an appropriate covariance function, inference method, training data ratio, and noise parameter. Using the simulation framework developed by Imken, the trained Gaussian Process model is utilized to predict safing events and recovery durations for NeMO and compared with the existing prediction methodologies. Thus, based on the frequency and outage time of a safing event predicted for a mission, it enables mission designers to make more informed decisions on tracking, safing recovery, and missed thrust requirements.

2. THEORY

The following section includes the statistical hypothesis tests utilized, parametric analyses, and goodness-of-fit criteria conducted with Weibull distributions, and predictive analytics using Gaussian process models. Note that in the discussion of the theory, the dataset referenced includes the time-between-events and the recovery durations for all missions. Based on certain mission classifiers that are defined in the next section, subsets of each of the full datasets are also considered.

2.1. Weibull Distribution

A Weibull distribution is commonly used in reliability analyses due to its flexibility in being able to model a dataset with just two parameters: the shape, β, and scale, θ. The shape parameter is a dimensionless, positive parameter and the scale parameter is in the units of time and also positive. Equation 1 shows the reliability function, Equation 2 shows the Weibull probability density function (PDF), and Equation 3 shows the cumulative distribution function (CDF) [8].

$$R(t) = \exp\left[-\left(\frac{t}{\theta}\right)^{\beta}\right] \tag{1}$$

$$f(t;\beta,\theta) = \frac{\beta}{\theta}\left(\frac{t}{\theta}\right)^{\beta-1} R(t), \quad \forall \quad t \geq 0 \tag{2}$$

$$F(t;\beta,\theta) = 1 - R(t), \quad \forall \quad t \geq 0 \tag{3}$$

A single Weibull distribution can only show a single trend from the dataset and may inaccurately model the time-between-events and recovery durations. A finite mixture distribution, which is a linear combination of multiple distributions, can be used to correct and better represent the data in some cases.

In this analysis, a combination of two Weibull distributions with weights for each distribution are considered. This is

done to understand if the data exhibited bi-modal behavior and is a better fit than a single Weibull distribution. This paper will henceforth reference the single Weibull distribution as the '1-Weibull' and the two Weibull mixture distribution as '2-Weibull'. Equation 4 shows the reliability function for the 2-Weibull distribution; the CDF remains the same as in Equation 3.

$$R(t) = (\alpha)\exp\left[-\left(\frac{t}{\theta_1}\right)^{\beta_1}\right] + (1-\alpha)\exp\left[-\left(\frac{t}{\theta_2}\right)^{\beta_2}\right] \tag{4}$$

where: $0 \leq \alpha \leq 1, \quad \theta_j > 0, \quad \beta_j > 0, \quad \text{all} \quad t \geq 0.$

2.2. Weibull Parameters using Maximum Likelihood Estimation

There are multiple ways to determine whether a certain scale and shape parameter of a Weibull distribution fits the data as best as possible. A Weibull plot is one that linearizes the axes such that the data fits the estimated Weibull reliability $\hat{R}(t)$, in a linear manner. Data aligned along the $\hat{R}(t)$ line in the [ln(t); ln(-ln(R(t)))] space is considered an appropriate fit for a Weibull distribution using this graphical estimation technique.

However, a more rigorous test that is able to deduce optimal parameters is the maximum likelihood estimation (MLE) methodology. The basic concept involves formulating a likelihood function and then finding parameter(s) that maximizes that likelihood function. Saleh and Castet define the likelihood function as,"the probability of obtaining or generating the observed data from the chosen parametric distribution" [8]. Equations 5 to 9 show the MLE setup for computing optimal scale and shape parameters for a 1 and 2 Weibull distribution. The full derivation is detailed in Chapter 3 of [8]. An abbreviated version is shown below.

Let $\boldsymbol{\theta}$ be the column vector of all parameters that need to be estimated using MLE; two parameters for 1-Weibull and five parameters for 2-Weibull. The likelihood function, $L(\boldsymbol{\theta})$, for a single Weibull distribution is formulated in Equation 5: where f is the PDF, R is the reliability function, t_i is the i^{th} safing event, δ_i is the i^{th} censoring value, and n is the number of safing events. In this analysis, although no censoring of the data is done, the aforementioned equations with censoring are still implemented; thus, δ is a column vector filled with ones.

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(t_i;\boldsymbol{\theta})^{\delta_i} R(t_i;\boldsymbol{\theta})^{1-\delta_i} \tag{5}$$

By transforming f and R into equivalent extreme value distributions, new PDFs and reliability functions can be formulated. For convenience, the natural logarithm of the likelihood function is taken to yield the log-likelihood equation, seen in 6.

$$l(\boldsymbol{\theta}) = l(u,b) = \ln L(\boldsymbol{\theta}) = -\left(\sum_{n=1}^{n}\delta_i\right)\ln b + \sum_{n=1}^{n}(\delta_i z_i - e^{z_i}) \tag{6}$$

where: $y_i = \ln t_i, \quad u = \ln\boldsymbol{\theta}, \quad b = \beta^{-1}, \quad z_i = (y_i - u)/b$

For the 2-Weibull distribution, a similar log-likelihood function can be developed. The PDF and reliability functions are transformed into extreme value distributions as seen in Equations 7 and 8 respectively where $\boldsymbol{\theta} = [u_1, b_1, u_2, b_2, \alpha]^T$.

$$f(y_i,\boldsymbol{\theta}) = (\alpha)f_1(y_i,u_1,b_1) + (1-\alpha)f_2(y_i,u_2,b_2) \tag{7}$$

$$R(y_i,\boldsymbol{\theta}) = (\alpha)R_1(y_i,u_1,b_1) + (1-\alpha)R_2(y_i,u_2,b_2) \tag{8}$$

where: $y_i = \ln t_i, u_j = \ln\theta_j, b_j = \beta_j^{-1}, z_i = (y_i - u)/b$. The log-likelihood equation is then defined as seen in Equation 9.

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n}[(\delta_i)\ln f(y_i,\boldsymbol{\theta}) + (1-\delta_i)\ln R(y_i,\boldsymbol{\theta})] \tag{9}$$

The optimal parameters, $\hat{\boldsymbol{\theta}}$, can be computed using traditional optimization methods. Maximizing $l(\boldsymbol{\theta})$, or equivalently minimizing $-l(\boldsymbol{\theta})$, is done using a quasi-Newtonian optimization algorithm − Broyden-Fletcher-Goldfarb-Shanno (BFGS), which does not require explicit gradient formulation. The built-in MATLAB function *fminunc* is able to perform this unconstrained minimization of the log-likelihood function. Certain convergence issues can arise such as finding local minima or not converging if the initial guess is in an unstable region. The initial parameters used in the optimizer are found using trial-and-error and best-judgment. Future methodologies could include more robust ways of computing initial parameters to improve upon convergence properties and finding global optimal solutions.

2.3. Goodness-of-Fit (GOF)

Once the optimal parameters are found for both the 1-Weibull and 2-Weibull distributions as described above, a criteria is necessary in order to evaluate whether either distribution is a good fit both in an absolute and relative manner. Two criteria are used to determine the goodness of the fit to the empirical data: Mean Square Error and Akaike Information Criteria. A goodness-of-fit metric is important to perform a parametric analysis because it describes how well that model fits the set of data in a statistically rigorous manner.

2.3.1. Mean Squared Error (MSE)—The MSE of a predictor $\hat{Y}$ is defined as the average of the square of errors/deviations. MSE is the second moment of an error and thus it captures the variance of that predictor plus the square of its bias [9]. In certain cases, even if the variance of a certain predictor is higher, the overall MSE may be lower due to a lower bias. Therefore, the MSE is chosen as a way to compare the goodness-of-fit rather than just the variance for the 1-Weibull and 2-Weibull distributions compared with the safing event database.

If $\hat{Y}$ is the estimated predictions from a Weibull distribution, and Y is the empirical safing event time between events and recovery times, then the MSE of the predictor or Weibull distribution is defined in Equation 10. The difference of $\hat{Y}$ and Y is commonly called the residual as it compares the predicted values and empirical values.

$$MSE(\hat{Y}) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \tag{10}$$

In a relative sense, if the 1-Weibull and 2-Weibull need to be compared against the dataset, a ratio of the two MSE values computed could provide insight into the relative strength of each regression. Thus the relative efficiency can be computed as shown in Equation 11.

$$MSE_{rel.eff.} = \frac{MSE(\hat{Y}_{1-Wbl})}{MSE(\hat{Y}_{2-Wbl})} \tag{11}$$

If this relative efficiency is greater than 1, then the denominator has a lower MSE value, and thus the 2-Weibull

distribution is a better predictor than the 1-Weibull. However, this methodology has a flaw in it. If a distribution is overfitted to the data, the MSE would falsely report that an over-fitted distribution is relatively better than one that is not. This motivated the search of another criteria for goodness-of-fit between the Weibull distributions.

2.3.2. Akaike Information Criteria (AIC)—In order to accurately judge the level of overfitting the model, a different criteria than MSE is needed. Initially, the Kolmogorov-Smirnov (KS) Test is considered. The one-sample KS test showed promise because it is a nonparametric test where the null hypothesis of the population CDF is equal to the hypothesized CDF. However, such nonparametric tests require independence between the empirical CDF and hypothesized CDF. If the hypothesized CDF is derived from the dataset, as the optimal Weibull parameters are derived using MLE, then there is no independence and thus the KS test would not be applied correctly [10]. Next, the Lilliefors test is considered; it is a two-sided goodness-of-fit test where the parameters of the null distribution are unknown and must be estimated. However, the test's formulation assumes either normal or exponential distributions. Developing this function for the Weibull distribution is nontrivial, and thus it is necessary to find other criteria. This analysis led to selecting the Akaike Information Criteria as a suitable criteria to compare the 1-Weibull and 2-Weibull distributions.

When estimating finite Weibull mixture distributions for reliability purposes, Elmahdy and Aboutahoun used the Akaikes Information Criteria (AIC) as a goodness-of-fit criterion [11]. AIC, founded on information theory, estimates the relative information lost in a given model that is derived from the data and trades off fit versus simplicity. AIC also only reports the relative quality of one model to another but gives no warning of absolute fit. AIC can be computed by Equation 12, where $L(\hat{\boldsymbol{\theta}})$ is the likelihood function and $l(\hat{\boldsymbol{\theta}})$ is the log-likelihood function, and k is the number of independently adjusted parameters that are being estimated or equivalently the number of entries in $\hat{\boldsymbol{\theta}}$ [12].

$$AIC = -2\ln L(\hat{\boldsymbol{\theta}}) + 2k = -2l(\hat{\boldsymbol{\theta}}) + 2k \tag{12}$$

Since the minimum log-likelihood value is already determined when computing the optimal Weibull parameters for the model, it is trivial to use the maximum value for both distributions in order to compute AIC. For the 1-Weibull model, $k = 2$ and for the 2-Weibull model, $k = 5$. Note that the AIC values are always positive since the negative of the log-likelihood value is used. When the sample size of the data, n, is small relative to the number of parameters, a corrected form of the AIC, seen in Equation 13, that adds the bias-correction term. The rule of thumb to use this corrected AIC (AICc) is when $n/k < 40$; in this paper, all AIC numbers reported are the AICc since the bias-correction term helps with the low sample size present in many of the subsets of the full data [11].

$$AICc = AIC + \frac{2k(k+1)}{n-k-1} \tag{13}$$

Recall that AIC is best at comparing relative models and not absolute if the PDF fits the data. The better model selected is the one with the lower AIC. Thus, the difference in AIC between the 1-Weibull and 2-Weibull distributions is computed, as seen in Equation 14.

$$\Delta_{AICc} = AICc_{1Wbl} - AICc_{2Wbl} \tag{14}$$

The relative strength of one model over the other is based on how large the difference is. The criteria for which model is better is shown in 15.

$$\Delta_{AICc} = \begin{cases} > 0 \rightarrow \text{2-Weibull distribution is a better fit} \\ = 0 \rightarrow \text{1 \& 2 Weibull distributions have same GOF} \\ < 0 \rightarrow \text{1-Weibull distribution is a better fit} \end{cases} \tag{15}$$

This methodology is better suited to compute goodness-of-fit since it can accurately gauge the significance of increasing the number of parameters and thus likelihood when fitting a model. By increasing the number of parameters used to fit data, the cost of overfitting is reflected in the AIC. The Bayesian Information Criterion (BIC) is also considered since the penalty on overfitting is larger, but determined not to be necessary since the difference in BICs computed give the same quantitative result as the difference in AICs.

2.4. Chi-Squares Hypothesis Test

Based on the mission classifier categories that the safing event database is split into, it is of interest to know whether two classifiers are statistically independent. The data is first divided up into rxc contingency tables such that the observed frequencies O_{ij} are quantified for two classifiers being compared. By taking the expected frequencies E_{ij} and observed frequencies, the Chi-Square statistic can be computed as seen in Equation 16. The *crosstab* MATLAB function helps automate this process by returning the table generated, associated labels, Chi-Square value, and the p-value.

$$\chi_0^2 = \sum_{i=1}^{row}\sum_{j=1}^{col}\frac{(O_{ij}-E_{ij})^2}{E_{ij}} \tag{16}$$

A hypothesis test can then be constructed for which a certain chi-squared threshold determines whether two classifiers are statistically independent based on the computed χ_0^2. Assuming a particular mission classifier A_i and B_i where A_i is not equal to B_i, the null and alternate hypothesis of independence can be stated as in Equation 17.

$$\begin{cases} \mathbf{H_0} : \text{Classifier } A_i \text{ is independent of } B_i \\ \mathbf{H_1} : \text{Classifier } A_i \text{ is dependent of } B_i \end{cases} \tag{17}$$

Combining the hypothesis test and χ_0^2 test statistic, a p-value can be computed for the mission classifiers between A_i and B_i. A smaller p-value indicates a lower probability of the null hypothesis being true. Thus, the smaller p-value means greater probability you reject the null hypothesis and a stronger conclusion that holds.

2.5. Gaussian Process Models for Regressions

Predictive modeling is the next natural step after analyzing a dataset in order to extract information and predict trends or patterns. Machine learning, evolved from computation learning theory in artificial intelligence, enables computers to automate learning and making predictions from data. Supervised learning algorithms, a specific class of machine learning, infer a mapping function based on user-provided input/output training data to predict new outputs given a certain input. Gaussian process models (GP model) is one type of supervised learning that uses nonparametric kernel-based probabilistic models to take a prior distribution for a given training dataset and obtain a posterior distribution for a set of new inputs [13] [14]. The total inoperability of a spacecraft due to safing events can be broken up into

two metrics; time-between-safing-events which acts as a frequency for the number of events possible, and the recovery duration for each safing event which acts as the amount of time the spacecraft is inoperable. Combining the two metrics will enable to quantify the total outage time or inoperability period of a spacecraft from safing events. In this paper, a GP model is used to predict new time between events (TBE) and recovery durations (RD) of safing events for a hypothetical new mission such as NeMO.

A few supervised learning algorithms are considered before settling upon the use of a Gaussian process model. These algorithms are typically divided into classification, clustering, or regression problems; predicting new TBEs and RDs is a classic regression problem. Thus, the algorithms considered include artificial neural networks, Gaussian process models, and regression trees. Since the regression tree and artificial neural network are eventually not considered the best algorithms for this type of problem (detailed in a future section), this section outlines the theory related to GP models.

One key assumption is that the arbitrary set of inputs, either TBEs or RDs, evaluated over a function is one sample from a multi-variate Gaussian distribution. In mathematical terms, this is defined using Bayes Theorem as seen in Equation 18 where tr refers to the training data [14].

$$P(Y|X, X_{tr}, Y_{tr}) \sim \mathcal{N}\big(Y_{tr}K(X_{tr}, X_{tr})^{-1}K(X_{tr}, X), \\ K(X, X) - K(X, X_{tr})K(X_{tr}, X_{tr})^{-1}K(X_{tr}, X)\big) \tag{18}$$

where $K(x, x^*)$ is the kernel function that maps an input from x to x^*. Noise is also added on the observed target values based on the confidence of the 'measurements' for safing event TBEs and RDs. Thus, another assumption made is that the noise processes have a Gaussian distribution for each observation n, seen in Equations 19 and 20, where β is a hyperparameter representing the precision of the noise. .

$$t_n = y_n + \epsilon_n \tag{19}$$

$$P(t_n|y_n) \sim \mathcal{N}(t_n|y_n, \beta^{-1}) \tag{20}$$

When training a GP model in order to find the optimal hyperparameters, the maximum likelihood function is computed to find the correlation length-scale parameter [13]. Rasmussen and Williams [14] extended this by incorporating a separate length-scale parameter for each input variable. While computing the optimal parameters, the relative importance of different inputs can be inferred from the data based on the value of the length-scale parameter. This methodology is called automatic relevance detection (ARD). Thus, it is possible to detect whether certain input variables will have a large or small effect on the predictive distribution because the 'weight' parameter is correlated with the normalized relative importance. The ARD framework is easily incorporated into various kernel functions. For safing events, this framework mathematically helps identify whether certain mission classifiers have a greater importance on predicting future safing event TBEs and RDs.

Rasmussen and Nickisch developed a MATLAB toolbox that enables users to train, predict, and deploy Gaussian process models [15]. A library of various covariance functions, mean functions, inference methods, and likelihood functions are available enabling easier implementation of a GP model [16]. There are two main functions that enable the use of Gaussian processes. One is the main *gp.m* function that is the main interface to the user for predicting data. The other is the *minimize* function that learns the hyperparameters by maximizing the log-marginal likelihood function. This is typically called the training portion. The implementation of a GP for the safing event database is done using the GPML toolbox.

3. Data Preparation & Statistical Analysis

3.1. Safing Event Dataset

The safe mode event database containing *Time-Between-Events* and *Recovery Durations* collected by Imken et al. is utilized in the same manner with the same set of assumptions: no cascading safing events, recovery durations from Galileo discarded, all events from the same population, etc. [3]. One important assumption is that the time-between-events and recovery durations for each safing event are assumed to be independent and identically distributed (iid). The rationale for this assumption is that it simplifies the analysis, although this may not be completely realistic if cascading safes are included. Generally, this assumption enables the use of classical statistical methods to analyze the dataset and subsets. Additionally, no data is assumed to be censored, in the context for parametric analyses.

Each mission and its associated safing events are categorized by four mission classifiers: Mission Class/Category, Mission Destination, Mission Duration, and SEP as seen in Table 6 in the Appendix. Each safing event is further classified by the safing event cause and by the location of the safing event in mission phase. The following list (including abbreviations) shows all the possibilities that a safing event can be classified under, and Figure 1 shows a histogram of the number of safing events for each classifier. The reason the number of valid safing events differ for time between events and recovery duration is due to the fact that certain events are omitted in one but not the other; the assumptions for omission are given by Imken et al. [3].

(1) **Mission Class**: Discovery, New Frontiers, Flagship;

(2) **Mission Destination**: Asteroid / Comet, Heliophysics / Exoplanet, Kuiper Belt Object, Moon, Mars, Saturn, Jupiter;

(3) **Mission Duration [years]**: 0-5, 5-10, 10-15, 15-20;

(4) **Solar Electric Propulsion**: Yes, No;

(5) **Safing Event Cause**: Environmental, Hardware, Operations, Software, Unknown; and

(6) **Safing Event Mission Phase**: Cruise-Primary, Cruise-Extended, Orbit-Primary, Orbit-Extended.

The motivation to categorize the data into such subsets is two-fold; one to enable the statistical and parametric analysis of these subsets in order to identify trends and independence, and two to use these as general inputs to the predictive model. A disadvantage of specializing the data in this manner is that it reduces the sample size for that mission classifier. By already having a limited dataset due to few interplanetary missions, certain analyses and predictions will have greater uncertainty.

Specific criteria are used to group missions into each classifier. For mission class, typical mission cost & mass are factored into categorizing missions. While some missions do not fall in the Discovery or New Frontier's program

that NASA currently conducts, certain missions, based on equivalent costs, are assumed to fall within that category. For mission destination, seven total destination categories are created based on the typical mission environment. All categories except three have more than one mission per destination category; the Moon, Saturn, and Kuiper Belt Object had only one mission's safing event for those categories. It was considered to combine two of those three categories, Saturn and Jupiter, into a new category such as the 'outer planets'; however, due to the different space environment faced at Jupiter versus Saturn, it was deemed to keep those separate. Although Heliophysics and Exoplanet missions seem mis-categorized, typically these mission remain in an heliocentric orbit. The mission duration is categorized based on their their launch date until the end-of-life or current date. They included both primary and extended mission phases. Three Mars landers and one failed Mars Orbiter only included the cruise phase of their mission as part of the safing event database [2]. For solar electric propulsion, the category simply stated whether SEP is part of the mission or not. For safing event cause and mission phase, the bins are determined based on the entry logs of safing events as determined by Imken [3].

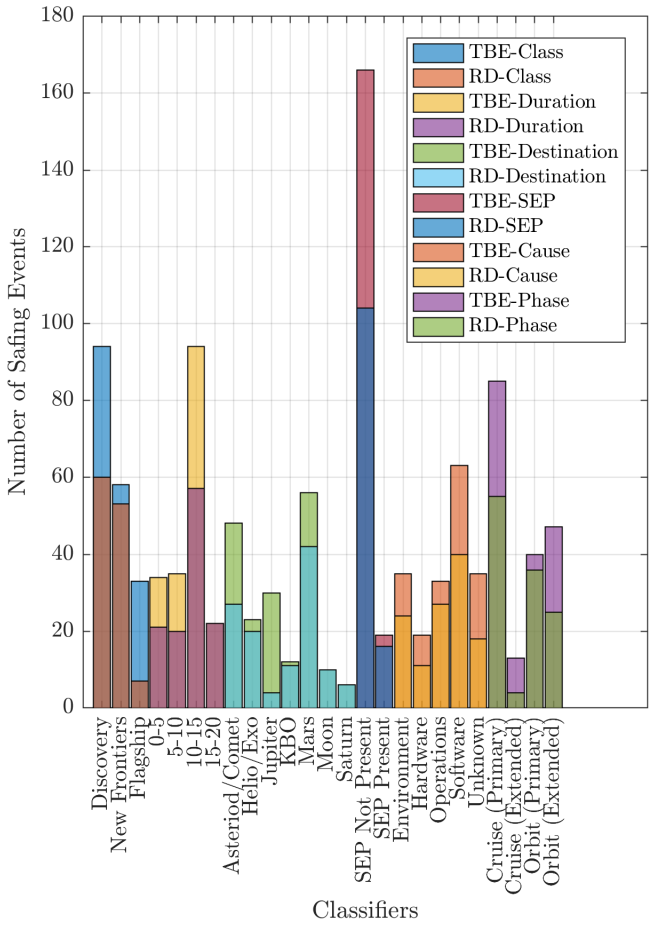

Figure 1. **Mission Classifier Safing Event Histogram**

3.2. Dataset Conversion for GP Model

In order for the mission classifier inputs to be correctly interpreted by the GPML, they must be converted from the categorical string inputs to numerical values. There are a total of 25 mission classifiers for which the categories must be encoded into a binary format. First, each mission classifier category is split up based on the number of mission classifiers. Since there is no ordinal relationship between each mission classifier in a category, the one-hot encoding methodology is applied. This is the case where a new binary variable is added for each unique value. Integer encoding is employed when sequential integers are applied to a particular category. By assuming a natural ordering between classifiers, poor performance or invalid results such as predictions between classifiers, resulting in a non-integer value could occur by utilizing integer encoding. Therefore, one-hot encoding is applied to each mission classifier category and then those binary numbers are concatenated together to form a 'chromosome' where all inputs are specified in a binary format. For example, the mission class category is encoded as seen in Table 1.

Table 1. **Mission Class One-Hot Binary Encoding**

Mission Class	Binary
Discovery	[1 0 0]
New Frontiers	[0 1 0]
Flagship	[0 0 1]

A similar encoding scheme is included for all other categories (mission duration, mission destination, safing event cause, and safing event mission phase). For the electric propulsion category, a single number is used to represent whether a mission had EP on-board or not: 1 or -1, respectively. Prediction performance may be better handled with a nonzero binary representation for only two categorical inputs. The mission elapsed percent (MEP) is also included as the last category as part of the chromosome for input purposes. This is a continuous, positive real number valued from 0 to 1 and thus did not need to be converted to binary. Concatenating each category's representation together, a total of 25 numbers (24 binary and 1 real-valued) represented the input space that is used as inputs to the GP model, shown in Equation 21.

$$\begin{aligned} &\text{GP input} = chromosone \\ &= [\text{Class}, \text{Destination}, \text{Duration}, \text{SEP}, \text{Cause}, \text{Phase}, \text{MEP}] \\ &= [1:3, 4:10, 11:14, 15, 16:20, 21:24, 25] \end{aligned} \tag{21}$$

3.3. Mission Classifier Independence Hypothesis Test

One way to understand the safing event database is to see if mission classifiers are statistically independent or dependent to one another. For prediction purposes, this will highlight the cross-correlation between two categories if two classifiers are dependent. Thus, based on the chi-squares hypothesis test formulation, a contingency table and the associated p-values for each classifier category are compared with one another. The hypothesis test is repeated for all permutations of each classifier, thus creating 30 valid p-values. Due to the commutative property, since category A_i being independent of category B_i is the same as B_i being independent of A_i, 15 unique and valid p-values are computed between each classifier as shown in Tables 7 and 8 in the Appendix.

Based on the relative p-values calculated, a confidence level of α = 1% = 0.01 is selected as the minimum confidence needed to make a decision on the hypothesis. Therefore, all conclusions on independence between two classifiers are made with a 99% confidence level. Recall that the null hypothesis is that two mission classifiers are assumed to be

[2] **Cruise phase of mission only**

independent; the alternate is that they are dependent of each other. If the p-value reported is greater than 0.01 then the test fails to reject the null hypothesis that two classifiers are independent; if the p-value is less than 0.01, then the null hypothesis that two classifier are independent is rejected. A rejection of a null hypothesis is statistically regarded as a strong conclusion; whereas, a failure to reject the null hypothesis is regarded as a weak conclusion. It is formulated as a failure to reject the null rather than as an acceptance of the alternate hypothesis, since no statistically significant conclusion can be made.

For the time between events dataset, the conclusions on the hypothesis are enumerated as follows:

(1) **Fail to reject the null hypothesis**: SEP & Duration, SEP & Cause, SEP & Phase, Cause & Phase

(2) **Reject the null hypothesis**: All other classifiers

A few interesting observations can be made from Table 7 and the hypothesis conclusions. First, mission destination is highly dependent with the mission class and mission duration categories since the p-values are very small (on the order of 10^{-20} and even smaller). This result suggests that the mission's destination is highly coupled with the class of spacecraft and how long it will operate. Those classifications combined could significantly dictate the time between safing events. The solar electric propulsion category concludes that with Duration, Cause, and Phase, there is not enough statistical significance that those categories are dependent. It is wrong to advance that conclusion for those categories being independent since it is a weak conclusion. However, between SEP and mission destination, there is a far stronger conclusion that those two classifier categories are dependent due to the small p-value. These results with SEP show an interesting trend that the destination plays the largest role in safing events compared with other categories. Another trend identified is that regardless of the safing event cause or what mission phase it is in, a SEP mission's predicted time between safing events will not be impacted by cause, duration, and phase.

For the recovery durations dataset, the results are enumerated as follows:

(1) **Fail to Reject the null hypothesis**: Cause & Class, Cause & Duration, Cause & Destination, Cause & SEP, Phase & Class, Phase & SEP

(2) **Reject the null hypothesis**: All other classifiers

Similar to TBE, for the recovery durations, the mission destination is highly dependent with mission class and mission duration. Mission destination is also coupled with SEP and mission phase, but association with the two is not as strong for recovery durations as evidence from larger p-values as compared with class and duration. The safing event cause category is dependent only with mission phase. SEC and destination have a p-value near 0.01, which border the conclusions made at the 99% confidence level. With a lower confidence, one could conclude those categories are dependent. This would align with the intuition since the recovery of SEP missions depends on location due to inherent complexity as well as round-trip-light-time which is not differentiated in this analysis. Finally, the mission phase and whether a mission has SEP and its class is agnostic to the recovery duration since failure times would not be coupled together.

4. PARAMETRIC ANALYSIS

On a reliability dataset, typically nonparametric and parametric analyses are performed to better understand the implications of the data. In the case for safing events, a parametric approach is taken due to the flexibility and convenience of modeling such events. A parametric model enables mission designers to easily implement safing event models in other studies as well as identify trends and patterns in the data.

4.1. Weibull Distributions

A parametric analysis of Time-Between-Safing-Events (TBE) and Recovery Duration (RD) for safing events is performed by modeling the data using the 1-Weibull and 2-Weibull mixture distributions. As used in Equations 1 to 4, each t value corresponds to either TBE or RD. Thus, there are four reliability functions formulated: two for when t = TBE and two for when t = RD.

Figures 2 and 3 show the CDF, Weibull probability plots, and optimal parameters for the 1-Weibull & 2-Weibull distributions for TBE and RD, respectively. The first and third subplot of each figure show the CDF; that is the cumulative probability that either a safing event will occur or if a safing event is completed. The maximum likelihood estimation (MLE) methodology outlined in the theory section is utilized to compute the optimal model parameters for each CDF. The TBE CDFs show that 400 days after the previous safing event or start of mission, there is a 80% probability using the 1-Weibull and a 78% probability using the 2-Weibull that the next safing will occur. Similarly, the RD CDFs show that after 72 hours of a spacecraft entering safe mode, that there is a 71.5% probability using the 1-Weibull and a 74.5% probability using the 2-Weibull that the recovery duration period will end. Since the 1-Weibull and 2-Weibull CDFs generally have similar predictions, certain criteria are explored in later sections to evaluate a preference between these CDFs.

The second and fourth subplots show probability plots for a 1-Weibull and 2-Weibull distribution respectively. Probability plots are used to graphically highlight how well data fits against each model. Since the Weibull distribution is linearized across its axes, if the data also is linear with the same slope, then the Weibull distribution is a good match. If there is curvature in the data away from the Weibull model line, then the probability plot indicates either a different distribution may fit better or a mixed distribution may be more ideal. Thus, for both 2-Weibull probability plots, the mixed distribution is better able to capture the curvature in the data from the first and second half due to the added degree of freedom. By looking at corresponding CDFs, from a graphical perspective, the 2-Weibull distributions fits the empirical CDF better.

Since implementation of the MLE methodology is done in MATLAB, validation is required to see if the optimal parameters outputed are truly optimal. Validation of the 1-Weibull and 2-Weibull MLE is conducted by using the Weibull++ software by Reliasoft Corporation. This software specializes in the analysis of reliability data; thus, it is chosen to validate the implemented MLE methodology with an industry standard software package. For the computed 1-Weibull distribution, the developed MATLAB implementation gave the same results compared to Weibull++ and to MATLAB's built-in function *wblfit*. For the 2-Weibull distribution, Table 2 shows the optimal parameters from the developed MATLAB implementation and the Weibull++ software using MLE

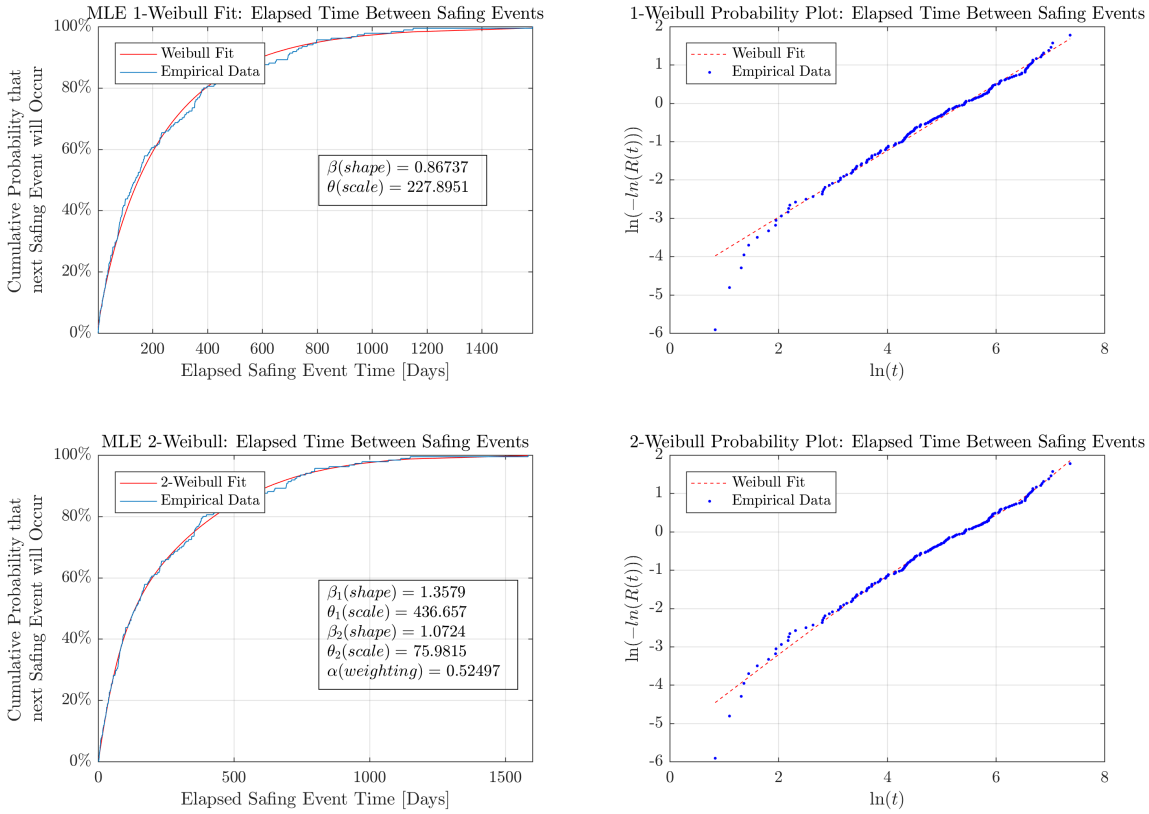

Figure 2. 1-Weibull & 2-Weibull Optimized Parameters, CDFs, and Probability Plots for Time-Between-Events

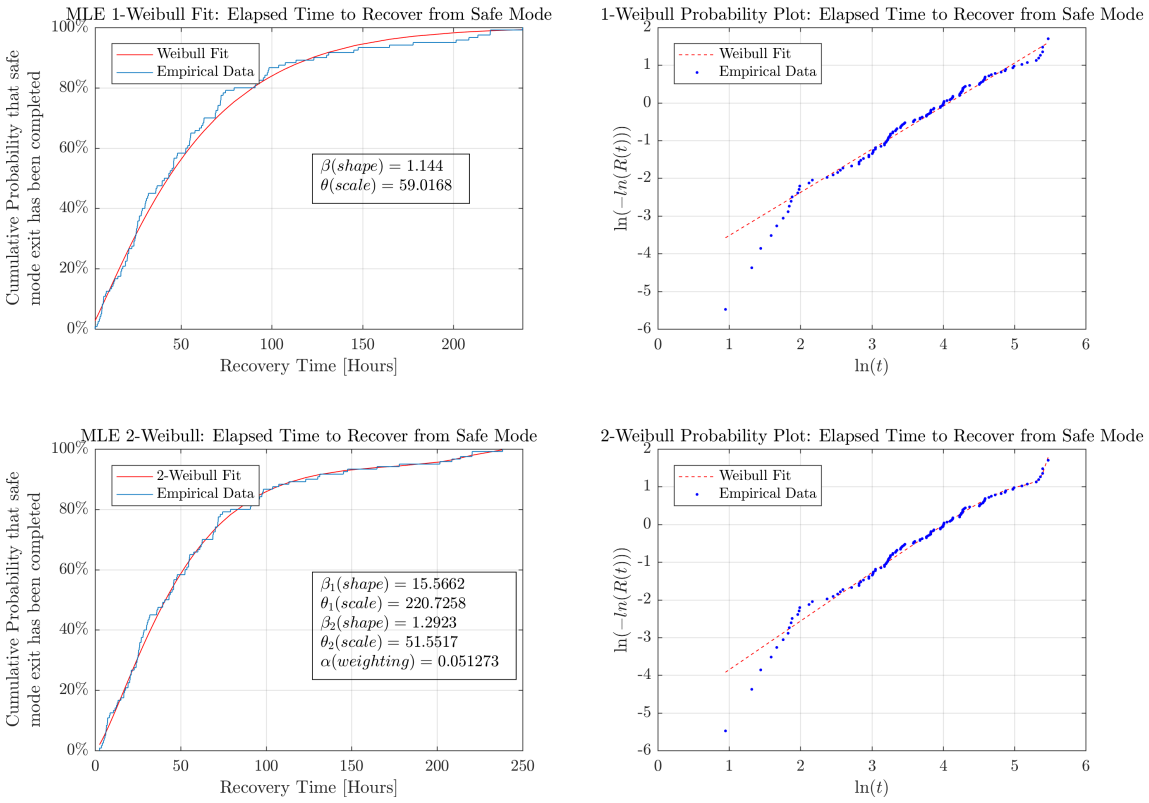

Figure 3. 1-Weibull & 2-Weibull Optimized Parameters, CDFs, and Probability Plots for Recovery Duration

methodology for both TBEs and RDs. The percent difference between each parameter is no greater than 5% for all except one parameter (θ_1 for TBE). Thus, the optimal Weibull distribution parameters for both the 1-Weibull and 2-Weibull are validated from the Weibull++ software which gives confidence in the results that the MATLAB implementation of maximizing a log-likelihood function is correct.

Table 2. 2-Weibull Optimization and Weibull++ Optimal Parameters using MLE

	Model	β_1	θ_1	β_2	θ_2	α
TBE	Matlab MLE	1.35240	434.03770	1.08370	79.79530	0.52820
	Weibull++	1.23565	391.07633	1.10943	67.36752	0.59849
RD	Matlab MLE	1.15460	148.00450	1.52820	47.54130	0.40869
	Weibull++	1.13074	142.80304	1.54750	47.33653	0.42769

In reliability engineering, the shape parameter β affects the failure rate as predicted from the reliability function $R(t)$. A $0 < \beta < 1$ implies a decreasing failure rate, allowing the function to model infant mortality, and a $\beta > 1$ implies an increasing failure rate, thus modeling wearout [4]. For the 1-Weibull TBE CDF, the shape parameter is less than 1 indicating that on a day-to-day basis, there is a lesser likelihood that a safing event will happen. For the 1-Weibull RD CDF, the shape parameter is greater than 1 indicating that the likelihood of recovering from safe mode after a given elapsed time increases with each passing day. While a mixed-Weibull fits the data better, insight into the behavior of the data becomes less transparent since the shape parameter seems to provide no such simple conclusions. The 2-Weibull distribution for TBE has shape parameters implying wearout for both modes of the dataset. The infant mortality found from the 1-Weibull is not captured in the 2-Weibull and thus such verdicts are inconclusive. The 2-Weibull distribution for RD implies that the larger shape parameter dominates the distribution such that there is an increasing convex failure rate. However, each of these preliminary conclusions is based off the assumption that by analyzing a mixture distribution, the shape parameter is able to provide conclusions regarding wareout or infant mortality.

Figures 2 and 3 only show the Weibull distributions applied to the entire TBE and RD datasets. One way to model certain categories of data better as well as analyze the trends is to take the subsets of data for both TBE and RD from the mission classifier categories and apply the same MLE methodology. For each mission classifier, 1-Weibull and 2-Weibull optimal parameters using MLE, CDF distributions, and probability plots are created. Although each associated figure is not shown in this paper for brevity, discussion about certain trends and a summary of the goodness-of-fit is shown in the following section.

4.2. Goodness-of-Fit

While the 1-Weibull and 2-Weibull distribution MLE methodology is validated to obtain optimal distribution parameters, the question whether a 2-Weibull mixture distribution truly models the data better requires further analysis. Thus, the motivation to calculate the mean square error (MSE) comes from wanting to quantify how well the Weibull fit estimated the empirical CDF. The residuals are computed from the difference of the each Weibull model (single or mixture) and nonparametric empirical CDF. The MSE helps determine the overall variance and bias of the Weibull distribution and the MSE_{eff} illustrates the ratio of the 1-Weibull MSE to the 2-Weibull MSE.

One major drawback of the MSE relates to over-fitted mixture models, which increase the number of parameters, a smaller MSE and a greater MSE ratio is computed as compared with a 1-Weibull's MSE. Thus, the Akaike Information Criteria (AIC) is better suited to compute goodness-of-fit criteria because the AIC value is penalized if more parameters are added to increase the maximum likelihood. A slight modification to the AIC that includes a bias-correction term (AIC_c) for small sample sizes is computed. The 1-Weibull and 2-Weibull CDF plot, probability plot, residual histogram, and MSE/AIC_c values are reported on three datasets of interest. The CDF plots also include the 95% confidence intervals for the empirical CDFs. This is important because if the Weibull distributions go outside of those bounds, then the confidence of the model decreases.

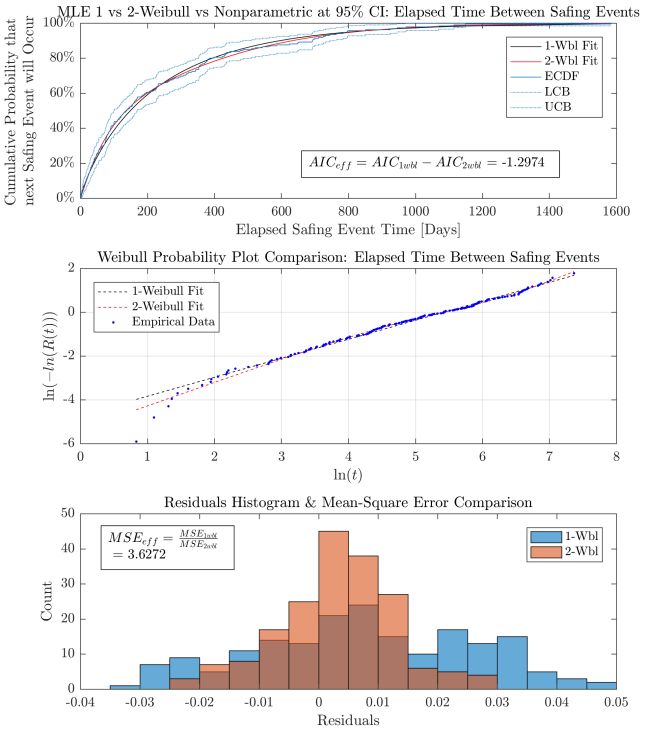

Figure 4. 1-Weibull & 2-Weibull Distributions Comparison for Time Between Events

From Figure 4 for the time-between-events, both the 1-Weibull and 2-Weibull CDFs stay within the upper confidence bound (UCB) and lower confidence bound (LCB). Although harder to tell from the CDF plots, the probability plots show how the slope of the 2-Weibull is lower initially than that of the 1-Weibull, but also curves upwards near the end of the dataset to better approximate it. The dispersion of the residual around the empirical CDF shows that the 2-Weibull distribution is a better fit than the 1-Weibull since the 2-Weibull MSE is 3.6 times better than that of the 1-Weibull. However, computation of the relative AIC shows that the 2-Weibull distribution overfits the TBE data. Even though the MSE and AIC values produce opposing conclusions, the the drawbacks of MSE hinder it from prevailing over the conclusion from AIC. Furthermore, this indicates that while the data may have some, bi-modal behavior in the data, it comes at a cost of overfitting the model and thus loses validity when choosing the 2-Weibull distribution for representing the TBE dataset.

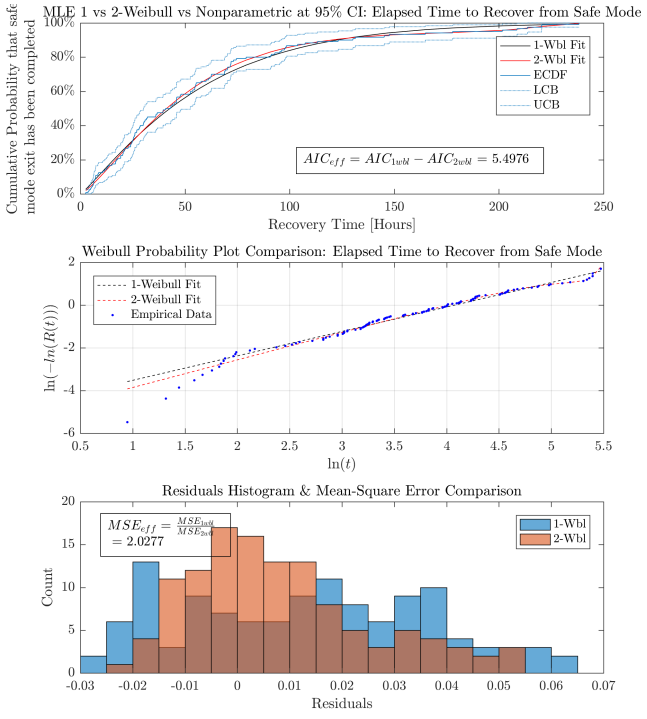

Figure 5. 1-Weibull & 2-Weibull Distributions Comparison for recovery durations

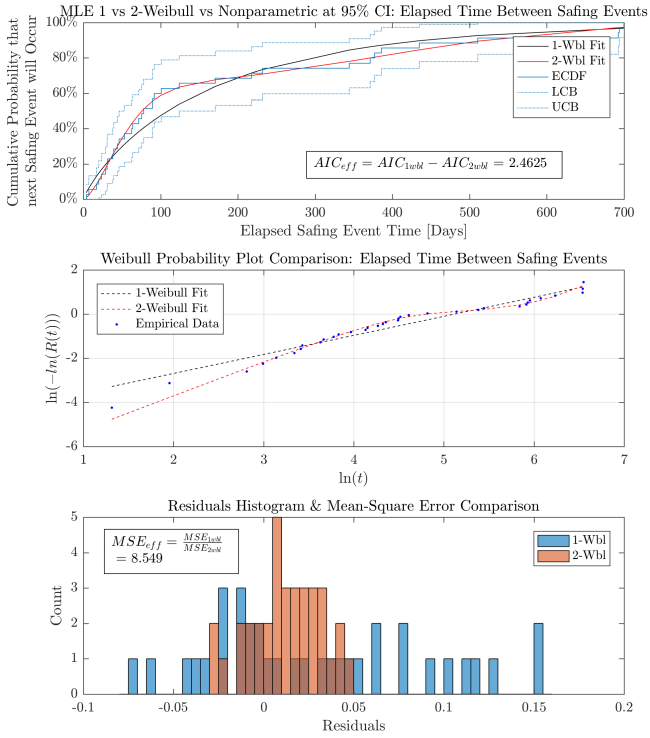

Figure 6. 1-Weibull & 2-Weibull Distributions Comparison for Time Between Events - Safing Event Cause - Unknown

From Figure 5 for the recovery duration dataset, both CDFs also stay within the 95% confidence bounds of the empirical dataset. The probability plot shows how the 2-Weibull is able to capture the bi-modal behavior in the data since it contains varying slopes to better fit the data. The residual subplot shows how the 2-Weibull has a more symmetric distribution of the residuals and smaller variance around 0 versus the 1-Weibull which has multiple peaks. From the goodness-of-fit computation, both the MSE ratio is greater than 1 and the AIC difference is positive, indicating that the 2-Weibull is a better representation of the recovery duration data and that it does not overfit the data. Thus, for future prediction purposes, mixed 2-Weibull distribution to predict recovery durations should yield more accurate results.

Specific trends about the data can be obtained from finding positive Δ_{AICc}. In the Safing Event Cause subset for TBEs, the unknown cause shown in Figure 6 shows a high degree of bi-modality in the data. This is evident from the high MSE ratio as well as positive AIC; furthermore, from inspection of the CDF, the slope in the probability plot is larger for the 2-Weibull than it is for the 1-Weibull. The reason for the bi-modal distribution may stem from two categories within the data. The first would increase the cumulative probability until the first 100 elapsed days are elapsed, and the cumulative probability would continue to increase, at a slower "rate" until 700 elapsed days due to the second category within the unknown safing event cause mission classifier. Further research within the dataset is needed to understand what contributes to the unknown cause classifier. It is clear from the AIC and probability plot that this mission classifier is better modeled by the 2-Weibull distribution.

Although all computed plots are not shown in this paper, the MSE and AIC for each mission classifier are tabulated in Table 3. For certain cases, the optimizer is not able to converge on the optimal parameters for the 2-Weibull distribution (highlighted as red). The limited subset of data for each mission classifier, shown in Figure 1 and used during optimization, is one major reason for convergence issues. The maximization of the log-likelihood is very sensitive to the initial guess; convergence problems typical with any optimization problem are encountered for some mission classifiers. For classifiers when the optimizer did converge for the 2-Weibull, the calculated MSE ratio and AIC difference values are reported. In most cases, the MSE ratio shows that the 2-Weibull distribution does a better job representing each data subset; however, only a select few mission classifiers show that for TBE and RD, the AIC favors the 2-Weibull. Since the AIC penalizes for overfitting the data, it makes sense that only for very large MSE values, that the Δ_{AICc} is positive. Furthermore, this result indicates that only a few of the mission classifiers should truly be represented by the 2-Weibull distribution and that their bi-modal behavior is inherent in their datasets. Overall, 26 total mission classifiers are analyzed for both TBE and RD. For the time-between-events data subsets, 20 successfully converged and gave results, 2 did not have enough data to converge, and 4 did not converge due to sensitivity to the initial guess. For the recovery duration data subsets, 14 are successful, 6 did not have enough data to converge, and 6 did not converge due to sensitivity to the initial guess.

Table 3. Weibull Convergence, MSE, and AIC Values for all Mission Classifiers. Red boxes indicate convergence for the 2-Weibull is unsuccessful. Yellow indicates that the 1-Weibull is a better fit and green boxes indicate the 2-Weibull is a better fit.

Mission Classifier	Category	2-Weibull TBE Convergence	TBE MSE	TBE AIC	2-Weibull RT Convergence	RT MSE	RT AIC
Time Between Safing Events / Recovery Time		Yes	3.63	-1.30	Yes	2.03	5.50
Mission Class	Discovery	Yes	3.25	-1.34	Yes	3.93	-3.45
	New Frontiers	No			Yes	0.56	-1.82
	Flagship	Yes	3.75	-3.94	No (not enough data)		
Mission Duration	0 to 5	Yes	1.62	-5.90	Yes	2.00	-6.55
	5 to 10	No			No		
	10 to 15	Yes	9.78	5.23	Yes	5.16	-3.12
	15 to 20	Yes	0.62	-7.23	No		
Mission Destination	Moon	No (not enough data)			No (not enough data)		
	Mars	Yes	1.96	-5.22	Yes	1.31	-6.56
	Jupiter	Yes	3.31	-1.94	No (not enough data)		
	Saturn	No (not enough data)			No (not enough data)		
	Kuiper Belt Objects	Yes	0.31	-13.16	No		
	Asteriods/Comets	Yes	4.49	3.42	Yes	3.22	-5.96
	Heliocentric/Exoplanet	Yes	1.63	-8.13	Yes	1.43	-8.40
Electric Propulsion	Yes	Yes	6.26	2.42	Yes	5.27	-3.82
	No	Yes	2.04	-3.38	No		
Safing Event Cause	Hardware	Yes	2.59	-2.39	No (not enough data)		
	Software	No			Yes	0.93	-5.12
	Operations	Yes	1.76	-3.02	No		
	Environment	Yes	3.73	-2.55	Yes	1.85	-6.94
	Unknown	Yes	8.55	2.46	Yes	1.62	-8.13
Safing Event Location	Cruise (Primary)	No			Yes	0.85	2.43
	Cruise (Extended)	Yes	1.53	-10.71	No (not enough data)		
	Orbit (Primary)	Yes	2.84	-5.08	Yes	3.44	-4.12
	Orbit (Extended)	Yes	1.62	-5.16	No		

5. Predictive Modeling

Analyzing the dataset through statistical and parametric techniques such as probability plots and chi-square hypothesis tests give a greater understanding of safing events and how various mission classifiers are correlated together. However, in the broad context of predictive analytics, one of the most crucial steps is modeling this dataset given multiple inputs that are specific to the problem. This enables a user to then predict time-between-events and recovery duration for a safing event and leverage those results for simulating operability.

Recall that various supervised learning models exist, each with its own advantages and limitations. For the application towards safing events, three regression algorithms are evaluated: artificial neural networks, Gaussian process models, and regression trees. Due to the number of mission classifiers and possible permutations of each category's classifier, a regression tree is too expansive to fully capture all possible scenarios. An artificial neural network (ANN) also showed promise since a variable number of neurons and hidden layers can be added to "learn" the system. By employing backwards propagation on the errors, the neural network's weights can be learned, thus allowing the network to weight each input accordingly based on training data. Many layers of neurons constitute a deep network, which would require a greater number of weights to be learned. In instances of data poverty, backpropagation techniques may miscalculate the neuron's weights. Neural networks typically succeed in situations where an abundant amount of data is present. This is not the case with approximately 180 and 120 valid safing events for time-between-events and recovery durations, respectively. A 16 node fully-connected neural network is implemented, but later disregarded due to the large mean error from the predicted testing data set. Thus, the validity and confidence of the predictions remained low even after employing techniques to improve performance including shuffling the data and using different activation functions. A Gaussian Process (GP) model showed promise as a regression algorithm due to its Bayesian framework rather than 'black-box' approach of neural networks.

A GP model can perform better with lower amounts of data because of the flexibility in adopting various functions in its computation. Rather than a deterministic output that an ANN produces, a GP model gives a mean and variance based on the confidence the model has for a new prediction. Therefore, by modeling time-between-events and recovery durations as a stochastic process that gives a posterior probability distribution, a GP model is chosen to learn and predict data for future safing events.

The GP model is first trained by taking the full data set and randomly dividing it up into training and testing data. Then,

it is initialized by setting the maximum number of conjugate gradient steps, the mean function, the covariance function, the likelihood and inference functions, and the initial values for the hyperparameters (covariance, mean, and likelihood). Selection of each of these values is very important, as it dictates how the GP model will learn the safing event data. The optimized hyperparameters are computed by minimizing the negative log-marginal-likelihood based off the training data. Using those hyperparameters, the testing data is provided into the GP model in order to compute the regression loss between the testing data and the predicted outputs. Through iteration and mathematical intuition, appropriate functions are selected, training data extracted, and hyperparameters initialized as to minimize the overall regression loss. Two separate GP models are developed: one for the time-between-events and another for recovery durations.

5.1. Training Gaussian Process Models

The training portion, which involves the selection of various parameters, data, and functions, of a GP model is the most important step to creating a successful predictive model. The criteria used to evaluate differences between models during training are the mean square regression losses. Two main figures of merit are computed: mean regression loss and variance regression loss. Minimizing the distance between the predicted mean value and the actual testing data is denoted as the mean regression error, as shown in Equation 22.

$$err_{mean} = Y_{TEST} - \mu_{TEST} \tag{22}$$

Obtaining the smallest variance away from the predicted mean is denoted as the variance regression error, as shown in Equation 23.

$$err_{var} = (\mu_{TEST} + \sigma \times \sqrt{Var_{TEST}}) - Y_{TEST} \tag{23}$$

Then, the mean square error is computed for both mean and variance errors as shown in Equation 24 where j is either the mean or variance, i is the testing data number, and N_{test} is the total number of testing data points evaluated.

$$MSE_j = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (err_j)^2 \tag{24}$$

Having a low mean regression loss indicates that the center of the predicted posterior distribution matches with the supplied testing output. A low variance regression loss indicates that the confidence of the GP model for a particular set of inputs is high.

During training there are a few fixed inputs and assumptions made to keep training computationally manageable. The maximum number of conjugate gradient steps during each minimization is limited to 10,000. For all cases, a standard deviation of two is observed because it encompasses 95.5% of all possible values in a normal distribution. A 2σ value is deemed sufficient for most scenarios for safing event predictions. Although certain training methodologies include a validation step to further tune the model while performing the minimization, no such validation is done. Since there is a limited dataset, having another chunk of the total data go towards validation, including testing and training, would reduce the amount of training data needed to sufficiently and accurately train the GP model.

5.1.1. Selection of Noise Parameter— In many instances, the data collected may not be perfectly captured and therefore have some uncertainty associated with its values. By including noise on the observed target values as seen in Equation 19, the uncertainty can be accounted for each time the spacecraft enters safe mode and how long it stays in safe mode. The noise processes are assumed to be normally distributed, from Equation 20. Models equivalent to the GP include the Kalman filter, which has extensively been used in guidance, navigation, and control applications. Kalman filters also capture noise through the use of a covariance matrix that accounts for the measurement uncertainty that sensors and actuators may give to the filter. Thus, the stochastic noise process in a GP can be thought of as a signal-to-noise parameter of the observations for the safing event data collected; if greater uncertainty existed for a particular measurement, a smaller SN ratio is used. In a GP model, the noise standard deviation parameter (SN) is incorporated into the likelihood function as a hyperparameter.

For the time-between-events GP model, SN values sampled ranged from 0.01 to 100 days. As the SN value reached towards 0.01, the GP model took the observations for when safing event happened as the 'truth' for the training points and thus, the model's mean connected all of the testing points. However on the opposite spectrum with a SN of 100, the testing points barely perturbed the prediction because the training values are assumed to be very unreliable. Since there is a good amount of confidence with the data collected and its sources, it is deemed that a noise standard deviation value for time-between-events would be 0.1 days.

For the recovery duration GP model, the same range of SN values are tested; however, the recovery durations is in units of hours. However, the confidence in the observations for when a spacecraft entered and exited is lower than for time-between-events. This is due to the fact that recovery periods documented include both subjective and objective values. Furthermore, exact durations down to the minute are typically not well documented and thus, the recovery durations usually are overestimates. Therefore, a noise standard deviation value of 1 hour is deemed appropriate for the level of confidence in the duration values collected.

5.1.2. Selection of Training Ratio— Next, selecting the amount of training data used versus the testing data needs to be determined. Using MATLAB's *dividerand* function, the dataset is randomly divided into three sets (training, testing, and validation) based on user-supplied ratios. For the GP model, it is assumed no validation dataset would be used; therefore, a training ratio is selected, and the remaining percentage would be used for testing. Possible training ratios considered are: 50%, 60%, 70%, 80%, 90%. Since *dividerand* randomly split the full dataset, 16 iterations per training ratio are computed as to determine what ratio would yield the lowest average and minimum MSE_{mean}, MSE_{var}, and negative log-marginal-likelihood (nlml) values. This is a brute-force methodology to remove the randomness associated with assigning different training data per iteration. 16 iterations are assumed to be sufficient enough for computational tractability purposes; however, more iterations could be included for future training purposes. Finding the smallest average values is more important because it showed greater consistency for that training ratio run across the 16 runs.

Thus, for the time-between-events GP model, a 70% training ratio is selected as the average MSE_{mean} and an average MSE_{var} are the lowest across different percentages. For the recovery duration GP model, an 80% training ratio is selected that had the lowest average MSE_{mean} and an av-

erage MSE_{var}. Since there are fewer valid data entries for the recovery duration dataset, it makes sense that a greater percentage of data is needed to accurately train the model.

5.1.3. Selection of Covariance Function—A covariance function is one of the core ways a prior distribution is determined. It describes the relationship between the function values of two points based on coordinate locations in an N-dimensional space. Since the training data are randomly selected, 16 iterations are again computed for each covariance function evaluated and the average and minimum MSE for the mean and variance are computed. Five possibilities are considered as viable covariance functions: squared exponential, Matern with $\nu = 1/2, 3/2, 5/2$, and the rational quadratic. Note that automatic relevance detection (ARD) is assumed for all covariance functions since it provided a means to understand the cross-correlation in the input space and appropriately weight each input (mission classifier) while training the model.

For the time-between-events GP model, the Matern covariance function with ν = 3/2 with ARD distance measure is selected. Although the computed MSE for the mean had a median value compared to other covariance functions, the MSE_{var} is the second lowest. Other covariance functions had their strengths in either a minimal MSE_{mean} or MSE_{var}, but the Matern 3/2 gave the greatest balance between minimizing mean and variance MSE errors. For the recovery duration GP model, the Matern covariance function with ν = 3/2 with ARD distance measure is also selected. The computed MSE_{mean} had a median value compared to other covariance functions, but the MSE_{var} is the lowest and thus selected. One reason why the Matern function also may be the optimal choice is because it contains the absolute exponential kernel, which may be able to better capture physical processes due to its finite differentiability [14].

5.1.4. Selection of Mean Function—A mean function typically helps specify where the expected posterior distribution's mean would lie. For both GP models, initially a mean function is not added as to not constrain the hyperparameters during minimization. However, constant and linear mean functions are also tested with varying initial condition. The results show that having a mean function gives lower mean squared errors. Thus, a constant mean function with an initial value of 200 days is set before the minimization for time-between-events. For recovery duration, the initial value for a constant mean function is set to 35 hours. A positive mean function created a non-symmetrical distribution around zero such that the probability of predicting a negative value would be far lower; essentially the posterior distribution is skewed towards positive values.

5.1.5. Selection of Likelihood & Inference Method— As stated by Rasmussen et al., "The likelihood function specifies the probability of the observations given the GP and hyperparameters. The inference methods specify how to compute with the model, i.e. how to infer the (approximate) posterior process, how to find hyperparameters, evaluate the log marginal likelihood, and how to make predictions" [15]. While all covariance and mean functions can be used without limitations, certain likelihood functions may only be used with particular inference methods. For a Gaussian likelihood function, an exact Gaussian inference method is used; however, for other likelihoods (e.g. Gamma, Weibull, etc.), a Laplace approximation to the posterior Gaussian process must be used. The likelihood functions that are evaluated included: Gaussian, Gamma, and Weibull. The Gamma and Weibull also has two possible inverse link functions, exponential and logistic, that are used to map from the GP to the mean intensity for a generalized linear model. Those two likelihoods are chosen over others to be evaluated because they apply to only strictly positive data, as is the case with the given time data.

For the time-between-events GP model, the Gaussian likelihood function had the second lowest MSE_{mean} and a median MSE_{var}. For the recovery duration GP model, the Gaussian likelihood function had the median MSE_{mean} and a low MSE_{var}. While the Weibull likelihood function had a lower MSE_{mean}, convergence for the algorithm is limited since the Gram matrix often became singular. The predictions from a Weibull likelihood would be invalid and thus the Gaussian likelihood and inference method is selected. Future work is necessary to adapt a Weibull likelihood function to properly converge; it may enable better predictions for positive values.

5.2. Fully Trained Gaussian Process Model

After the appropriate selection of each of the parameters and functions as discussed, a summary is shown for each GP model in Table 4. This table also includes the performance metrics that are computed with the particular testing data. While the lowest errors are chosen when selecting parameters during training, the performance metrics still illustrate that there is a significant amount of error in prediction. This is due to a number of factors such as a limited dataset, refinement in mission classifier definition, and the various assumptions made on the dataset. Moreover, these parameters are in no means the optimal configuration for predicting safing events; this is a preliminary result to establish the framework necessary to use GP models for prediction of time-between-events and recovery durations. Further studies focusing on training the GP models will be required to further reduce the mean square errors and negative log-likelihood.

Table 4. GP Model Summary

Parameter / Function	Time-Between-Events	Recovery Duration
Noise Parameter	0.1 days	1 hours
Training Ratio	70%	80%
Covariance Function	Maternard: $\nu = 3/2$	Maternard: $\nu = 3/2$
Mean Function	Constant: 200 days (initial)	Constant: 35 hours (initial)
Likelihood Function	Gauss	Gauss
Inference Function	Gaussian	Gaussian
MSE_{mean}	69365	1261
MSE_{var}	202028	12869
nlml	865	514

Once a model is trained, plots are generated to show how well the testing data is predicted by the GP model. A discrete number of testing points are evaluated by the GP model, which represented the $1 - TrainingRatio$ of the full dataset. The x-axis shows those training points numerically ordered on a linear scale; however, each point is actually a multi-dimensional representation of the input space (7 categorical/25 binary inputs). The rise and fall to the mean line shows how the GP model reacts to changes in particular inputs. The 2σ boundary shows the tail-end of the normal distribution centered around the mean; if the boundary is smaller, then the model has greater confidence in its prediction since it may have seen such testing data during training. Also, since the 2σ boundary predicts 95.5% of all data when it is normally distributed, it is possible certain TBE or RD testing points would lie outside of that boundary.

The outputs of GP model at a particular testing data point are the predicted mean and variance. In order to make a

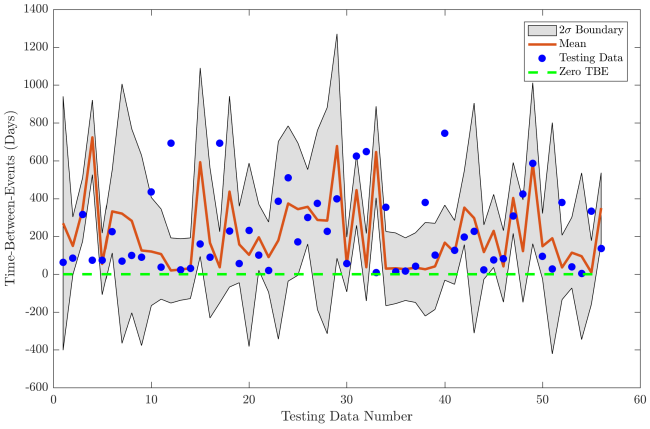

Figure 7. Time-Between-Events GP Model using Testing Data

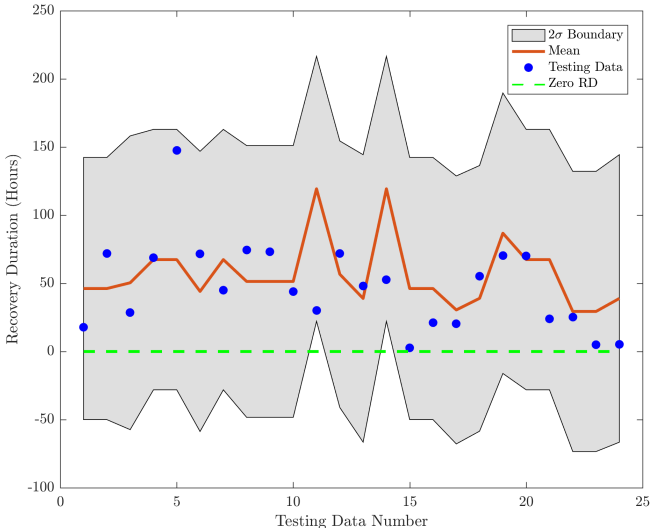

Figure 8. Recovery Duration GP Model using Testing Data

prediction, a single random value from a normal distribution using the computed mean and variance is generated. Since the testing data y-axis is in units of time, it is not realistic to predict negative times. If the 2σ boundary can be negative and a prediction made is negative, then new predictions are made until a positive value is obtained. While this a methodology may invalidate one sided tail of the distribution, the likelihood of obtaining a negative value remains low because the distribution is skewed towards positive values with a positive mean. Better model training is necessary to tackle this drawback of the currently trained GP model to eliminate prediction of negative numbers.

Figure 7 shows a plot of training dataset used for time-between-safing-events comparing the actual output versus the predicted. The mean line significantly varies between 150 - 600 days elapsed between events as it is perturbed by the inputs the testing data contains. For certain test data, the variance grows and shrinks based on whether or not the GP model can make an accurate prediction based on its training data and covariance weights. There are a few points that lie outside of the 2σ boundary, but as mentioned, two standard deviations only contain 95.5% of all data.

Figure 8 shows a plot of training dataset used for recovery duration for safing events. The observed mean value ranges from 40 hours up to 130 hours as the testing data changes the predictions of the model. Note that for a few testing data points, the recovery duration is close to zero, and the mean predictions also shift closer to those values. The RD model as compared with the TBE model predicts fewer changes to the mean but with a greater relative variance for each point.

6. Model Deployment for Predictions

6.1. Deploying the Model for NeMO

The final step in the domain of predictive analytics is to deploy the model for an actual scenario. Using the Next Mars Orbiter (NeMO) mission concept as a case study, the trained GP models are used to predict how long safing events would occur and how much time would pass between each event. Using the established mission classifiers, certain inputs are fixed for NeMO while others are time-varying based on the mission elapsed percent. This simulation will help quantify the impacts of safing events through the use of a more sophisticated model that uses various mission inputs in order to best predict time-between-safing-events and recovery durations. Furthermore, the results of the simulation will help better quantify mission inoperability rates for various prediction models, and shape requirements and system margins for a particular mission.

By leveraging the simulation work done by Imken et al., the trained GP prediction model framework is incorporated into the existing simulation shown in Figure 9. The same set of simulation parameters are used to generate new results and compare with a Weibull-based pseudo-random number generator. Figure 10 shows the GP model framework developed such that it would be very easy to incorporate into the existing simulation. The blue boxes in Figure 9 show where the GP model framework is incorporated into the full mission simulation in a "plug-and-play" manner. From the overall simulation to the GP model, the current mission elapsed percent (MEP) for that particular iteration only needs to be passed as an input. Then, using the MEP and a few other fixed inputs, the GP model's categorical inputs are created. Using the one-hot encoding scheme described earlier, the conversion from categorical to binary inputs is made. Once the training of the particular GP model (whether it is is for TBE or RD) is done, the training data and optimized hyperparameters are used to generate a prediction with a certain mean and variance. From that, a time is randomly generated using the computed mean and variance from a normal distribution (*normrnd* in MATLAB). Since the GP model is not bounded to be strictly positive, it is possible to obtain negative time values; thus the normal distribution is re-sampled until a positive value is obtained. Although a portion of the distribution to obtain a valid sample gets smaller, due to the skewed distribution computed, the likelihood is smaller than a traditional Gaussian distribution.

The seven categorical inputs that the GP model requires are listed in the inputs parallelogram in Figure 10. The first four inputs are fixed and constant based on the candidate mission that is be simulated. For the case of NeMO, the inputs are as follows assuming a total mission length of 6 years:

(1) **Mission Class**: New Frontiers;
(2) **Mission Destination**: Mars;

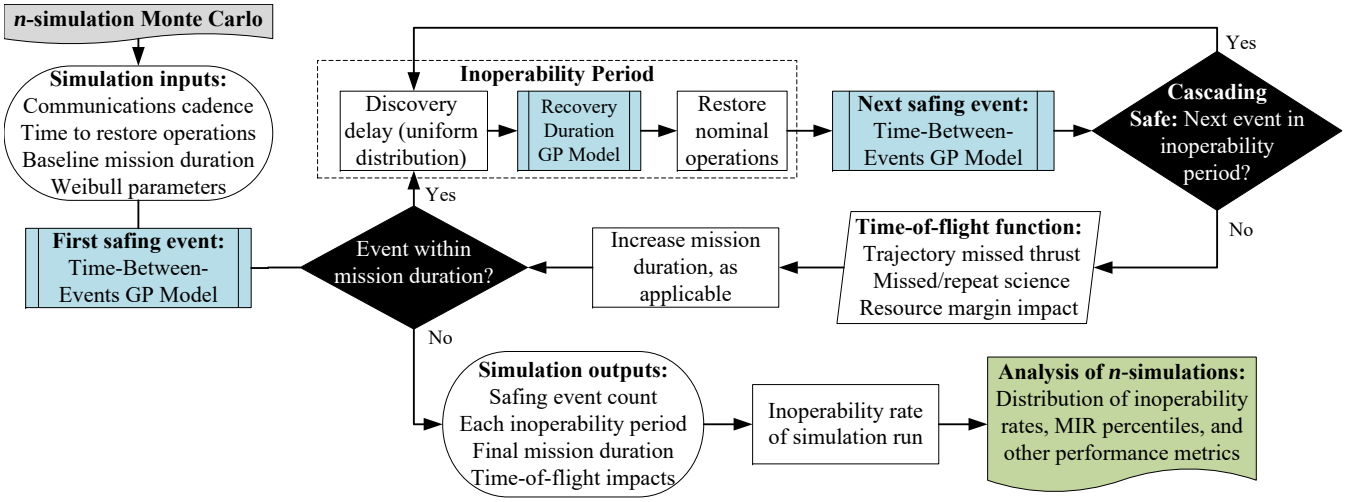

Figure 9. Safing Event Monte Carlo Simulation Block Diagram. Blue boxes indicate subprocesses, green is main simulation output, black boxes are decisions, and gray is the simulation start

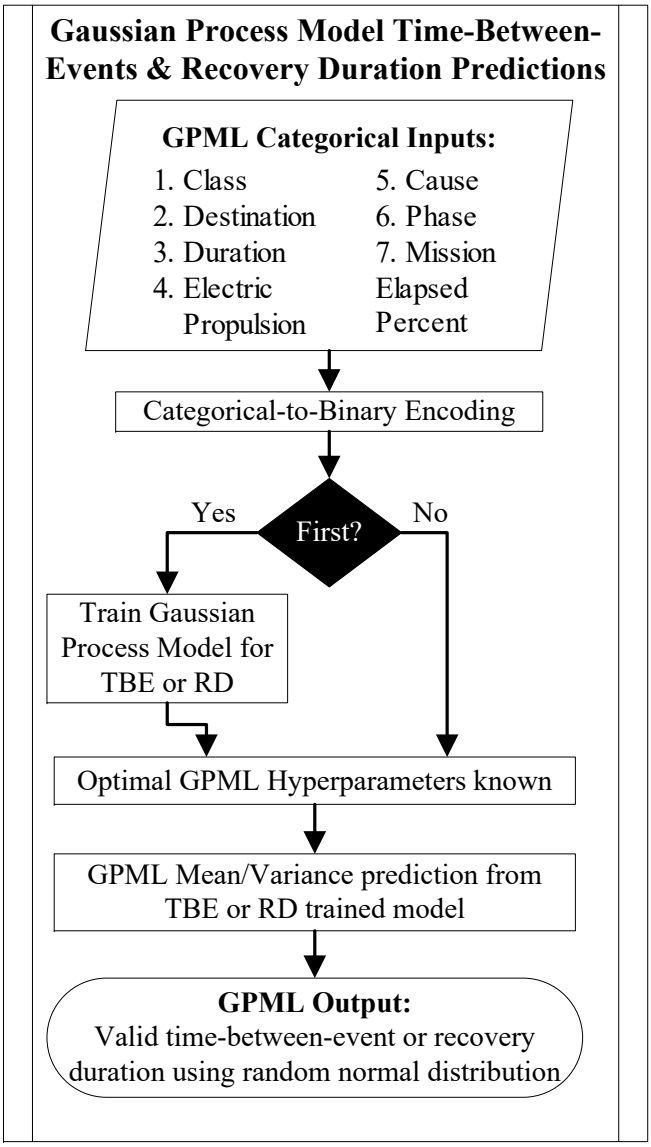

Figure 10. Gaussian Process Model Block Diagram

(3) **Mission Duration [years]**: 5 - 10; and
(4) **Solar Electric Propulsion**: Yes.

The mission elapsed percent is a real-valued positive number from 0 to 1 that gets passed from the main simulation to the GP model framework. Safing event cause and safing event mission phase are a function of the mission elapsed percent. The safing event mission phase is also a fixed based user-supplied mission phase. For the case study of simulating NeMO, no extended phases are incorporated. The primary cruise phase lasts 2/3 of the first year and during the remaining mission length, NeMO is in the primary orbit phase. These parameters are easily tunable for each mission and more complex logic to determine mission phase can be applied.

Safing event cause is also a function of mission elapsed percentage. A preliminary model is developed to predict causes. By using a kernel density estimator to smooth the histogram for safing event causes, a smooth, continuous probability curve is generated for each of the causes such that they sum to one. The bandwidth is a tuning parameter that controls the smoothness of the resulting density curve; through multiple iterations, a bandwidth of 0.2 was determined graphically as the 'best' fit. From the kernel-smoothed data, a cumulative weight distribution consisting of the five safing event causes for each mission elapsed percent is constructed. Then, from a uniform random distribution, one of the causes based on the weighted distribution is selected at that particular mission elapsed percentage.

Since the GP model framework only requires the mission elapsed percent to be passed into it, 25 MEP values are randomly selected from a uniform distribution and time-between-events and recovery durations are predicted. Figures 11 and 12 show the posterior Gaussian distributions for each MEP; this includes the mean and variance computed by the GP model. For the TBE results, it is interesting to see how the mean changes based on the mission elapsed percent. There is no clear trend that for NeMO, time-between-safing-events increase or decrease as a function of the mission time, but there are periods where the mean predictions may be higher or lower. The volatility in the predictions is due to the type of training data the model received and how the weights are formulated. For the RD results, there is far less change in

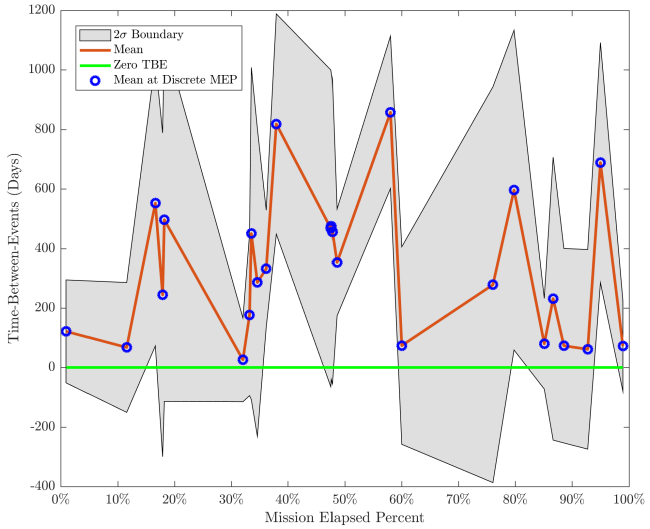

Figure 11. NeMO Sample Predictions using Trained GP Model for Time-Between-Events

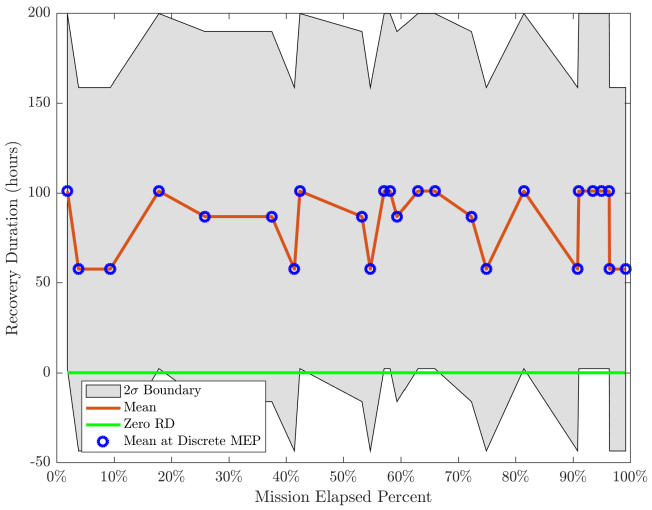

Figure 12. NeMO Sample Predictions using Trained GP Model for Recovery Durations

the mean values as a function of the mission elapsed percent. It seems that there are really two modes to the mean value where most values are predicted within: 55 and 101 hours. Most of the 2σ boundaries are also predicted to be positive, while only certain tail ends are below the threshold. Recall that any negative values predicted are not considered, and are re-predicted from a normal distribution with the given mean and variance.

6.2. Monte Carlo Simulation Results

Once the trained GP models for TBE and RD are implemented into the simulation framework, a safing event Monte Carlo simulation for NeMO is conducted for 1 million runs. The same set of assumptions that Imken et al. used in the simulation such as Deep Space Network DSN) pass cadence, pass length, time to restore to nominal operations, and time-of-flight function increase are retained. The time-of-flight function increase, relative to inoperability period is assumed to be zero; thus there is no increase with the total mission length due to additional safing events.

This Monte Carlo simulation is run three separate times, each with a different predictive model. The first model used is the 1-Weibull distributions that are discussed by Imken [3]. The next model included the use of the 1-Weibull distribution for the time-between-events and the 2-Weibull distribution for the recovery durations. This selection is based on the AICc analysis that determined for the full RD dataset, a 2-Weibull is a better predictor without overfitting the data. The final model is the use of the two Gaussian Process models that are developed and trained in this paper.

The results from each Monte Carlo simulation on mission inoperability rates for the three prediction models are listed in Table 5. Assuming a 99.7% likelihood, the total number of safing events for the GP model compared to two Weibull distribution models decreases from 19 to 11 events. Furthermore, the whole distribution predicted half as many safing events by the GP model than the Weibull models. This result for the sharp decrease in number of safing events is attributed to the fact that the time-between-events prediction from the GP model is larger than a similar result from the inverse Weibull distribution. Due to the TBE training data utilized as well as the covariance optimal weights, the mean predictions for the Weibull distribution models are lower than most of the mean predictions by the GP model for mission elapsed percentages. For both Weibull distribution models of a TBE, the mean is computed as 245 days between each event and the mean TBE predicted by the GP model ranges from 50 to 700 days as seen in Figure 11. Thus, the differences in the models lend to the significant difference in the predicted number of safing events.

The predictions for each recovery duration are larger for the GP model than predicted by both Weibull distribution models. Again, in this case for the Weibull distribution models, the mean recovery duration predicted by the Weibull models is approximately 56 hours compared with the observed 55 and 101 hours from the GP RD model as seen in Figure 12. Thus, the average recovery duration for the entire mission duration predicted by a GP model is higher than what is predicted by the Weibull distribution models. Since the mixture Weibull model uses a 2-Weibull for the recovery duration as opposed to the 1-Weibull for the single Weibull model, the outage times predicted vary. The mean recovery duration predicted for the single Weibull of 56.25 hours is slightly larger than predicted by the mixture model of 56.17 hours. Thus, it makes sense at the 99.7 percentile, that each outage time predicted by the mixture model is slightly lower.

Combining the total number of events and each outage time, the total outage time predicted by each model is around 80 days for the NeMO mission concept. The GP model predicts a slightly higher total outage time than both Weibull models, and the mixture Weibull model predicts a day higher total outage time during the mission than the single Weibull model. Thus, it makes sense that the mission inoperability rates (MIR) computed for the three models fall in that order: the GP model predicts the highest value, the mixture model predicts the median MIR, and the single Weibull predicts the lowest. This trend is highlighted in Figure 13 where the general shape for the likelihood of achieving a maximum inoperability rate is comparable for all three predictive models. Each maximum MIR value with a 99.7% probability is within the same percentage point. The maximum differences in MIR values corresponds to 0.15%, or equivalently for a 6 year mission, about 3.2 days of extra inoperability predicted by the GP model compared with the single Weibull model. This is corresponds to 12 hours extra per year and from an operations

Table 5. NeMO Mission Inoperability Rates, Outage Times, and Number of Safing Events for a 1 million run Monte Carlo Simulation for 3 Predictive Models

Metric	Units	Single Weibull Distribution Model	Mixture Weibull Distribution Model	Gaussian Process Model
99.7 Percentile: MIR	%	3.74	3.77	3.89
95 Percentile: MIR	%	2.78	2.83	2.9
99.7: Total Outage Time	days	81.91	82.63	85.15
99.7: Each Outage Time	days	12.9	12.57	17.71
99.7: Total Safing Events	#	19	19	11
99.7: Max Time of Flight	years	6	6	6
Max: Total Outage Time	days	126.86	126.79	128.4
Max: Each Outage Time	days	27.27	39.51	33.66
Max: Total Safing Events	#	30	29	16
Max: Max Time of Flight	years	6	6	6

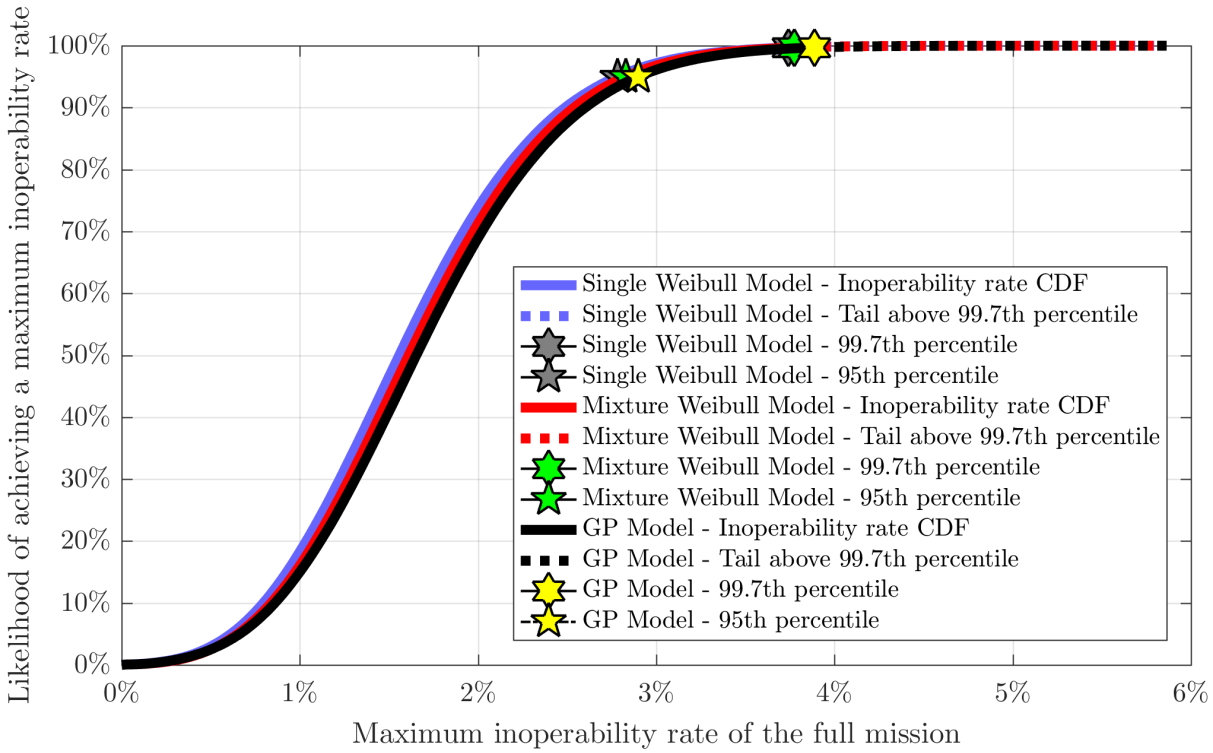

Figure 13. Distributions of Likelihood of Realizing a Maximum Inoperability Rate for NeMO by Comparing 3 Predictive Models

perspective, it could translate to one extra shift during one safing event.

There are a few implications on the design, margins, and requirements for the Next Mars Orbiter mission concept that the safing event predictive models provide in order to reduce risk. First, the GP model predicts that recovery times for NeMO would be longer when various mission inputs are factored in. More time spent recovering the spacecraft out of safe mode means a longer percentage of time that the mission is inoperable. This may motivate the development of greater autonomy on-board NeMO such that the spacecraft bus may be able to better diagnose certain events and provide more informative health data to ground operators. It would shape the requirements on NeMO to include an increased fault checking capability and/or better data management on-board. Additionally, the pass cadence assumptions for this simulation is one DSN pass every 3 days. The maximum outage time predicted to 3σ is predicted to be 17 days; an increase in the pass cadence could decrease that outage time. Moreover, the recovery urgency based on mission risk posture per safing event could increase such that the recovery period can shorten. While the cost to the mission, from DSN time, personnel, and other resources, would increase, the resulting increased operability of the spacecraft could be worth it for the mission's success. Furthermore, new requirements could be placed on the operations team such that greater confidence and faster response time dealing with fault scenarios are implemented.

Another impact from the overall increase in MIR predicted by the GP model is the missed thrust periods. Currently, there is no time-of-flight increase implemented in the simulation; however, a 3.9% mission inoperability would affect when the mission reaches its destination. Moreover, the consequence of missing thrust maneuvers during certain segments of the trajectory may significantly lengthen the mission. Those missed thrust periods may correspond to correlations greater

than 1:1 for each period. Extensions to the mission due to missing critical thrusting periods could significantly influence how margins are computed for a low-thrust mission. An increase in propellant margin would impact other margins such as mass and power, which would influence the spacecraft design considerably. In order to reduce the maximum inoperability predicted, spacecraft and operational capabilities may need to increase for the Next Mars Orbiter mission concept.

The mission inoperability rates, number of safing events, and outage times presented are mission specific to NeMO; as the inputs are changed for new missions, the results would also vary. Thus, it is up to the user to choose which model based on the given set of assumptions to predict safing events. To accommodate multiple mission inputs, the GP model enables users to factor the predictions made for the frequency and recovery duration of a safing event. Through the simulation, mission designers would be able to quantify the likelihood of realizing the worst-case inoperability rates, and make design and operational decisions based on the results.

7. RECOMMENDATIONS & FUTURE WORK

The assumptions, analyses, and modeling reported in this paper provide a methodology for future mission planners looking to predict safing events for a certain mission architecture. First, the user must decide what set of assumptions placed on the dataset and simplifications are acceptable for prediction purposes. Next, a safing event process model such as the 1-Weibull distribution model, Mixed Weibull distribution model, or Gaussian process model for safing event predictions is selected. When utilizing the GP model, the user must be aware of its 'black-box' nature that occurs during the training process and that re-training may be necessary. While the GP model developed in this paper created a safing event prediction model based on various mission inputs, improvements during training and tackling some of the assumptions can still be made.

One of the first set of simplifications is that subsets of the dataset are created using commonly defined mission classifiers. In order to rigorously find how to split the safing time-between-events and recovery durations based on the inherent divisions within the data rather than 'arbitrary' categories, more historical mission data statistical analysis is needed. Methodologies such as classification or hypothesis tests could be useful in finding these natural boundaries in the dataset. These new categories may lend to better predictions since the weights that the GP model learns would have a more statistically significant backing.

Training the GP model is the most important step to effectively utilizing a GP as a predictive model. The current method of computing the mean square error for deviations from the mean and variance estimates for the testing data is a good preliminary method. However, other metrics such as mean absolute error, sum absolute error, and others can be used for evaluation purposes. Furthermore, cross-validation is another methodology during training that allows to evaluate performance on a portion of the data that is not training and testing. Subject matter experts in supervised learning algorithms could lend guidance on the selection of the noise parameters and relevant functions (e.g. covariance, likelihood, etc.) for the GP model. More intuition from the mathematical theory is needed on the selection of certain functions.

Based on the trained GP model from this paper, negative outputs are possible within predictions for TBE and RD. One facet of this could be due to the inherent assumption that given the multi-dimensional input space of mission classifiers, the posterior distribution is a Gaussian distribution. The implementation of a Weibull likelihood in the GP model is one possible way to predict non-negative values. Another possibility is constructing a new optimization problem for a GP if the Gaussian distribution assumption holds such that a positive data constraint is applied.

While generating 2-Weibull distributions is successful for the full dataset and most of the subsets, there are instances where the optimizer did not converge due to a lack of data. While implementing the optimization algorithm, it is observed that the convergence is sensitive to the initial value. Thus, equations or shifting of the function may be necessary to get a rough estimate for starting values. Commercial software packages such as Weibull++ also exist that could be more robust during optimization.

One way that inference about a population from a dataset can be accomplished is by employing bootstrapping - sampling with replacement. Optimal Weibull parameters could be obtained for subsets of data by re-sampling the given subset and computing average parameters that would yield estimates to the true probability distribution. Bootstrapping could also be used in GP model estimation as a means to increase the size of the training, validation, and testing datasets. Consideration must be given when employing bootstrapping on the posterior distribution the model creates.

While only parametric analyses are considered in this paper, nonparametric analysis techniques such as the Kaplan-Meir estimator can be considered to understand the true nature of the data. One key aspect of nonparametric models is the censoring of data - when failure data is incomplete. When a mission reaches the end-of-life, one could apply right-censoring since that time should not be modeled as another safing event and would be stochastic across many missions. While initial thoughts were formulated, no sufficient conclusions are made on how best to censor data. The parametric framework developed to compute 1-Weibull and 2-Weibull distributions has censoring included in the formulation; thus, it should be easy to implement and obtain new Weibull distribution estimates.

Finally, regardless of how accurate a prediction model is developed, it is still limited by the data by which it is defined. As more interplanetary mission safing events occur, it is imperative to continue to collect data and store this additional information in the database. Then, when a 'significant' amount of data is added, re-training of the models may be useful to incorporate new information and re-weight accordingly.

8. CONCLUSION

Safe mode is an operational state that occurs when a space mission experiences anomalies and failures prompting the mission to execute actions that decrease further risk and operate only essential components. Missions that utilize solar electric propulsion such as the NeMO concept and Psyche have a need to more accurately model safing events since continuous operations are vital for their missions. Building on the work done by Imken et al., this paper statistically explores some of the intricacies of the existing interplanetary

safing event dataset, creates different models for prediction, and uses the NeMO concept as a case study for model deployment.

With the collection of time-between-safing-events and recovery durations for safing events, subsets of the dataset are created based on common mission classifiers. While these classifiers are made using categories that would be reasonable for a mission planner, future work could involve creating categories based on statistical significance between the data. A Chi-Squared hypothesis test reveals the degree of independence between each created mission classifier. Most classifiers are significantly dependent between each other. The computed p-values of safing event cause and whether a mission had solar electric propulsion indicates that those two classifiers are more independent from the rest of the subsets than other mission classifiers.

Parametric modeling of the safing event database is accomplished through the use of single Weibull distributions and mixtures of two-Weibull distributions. Using the maximum likelihood estimation algorithms, optimal Weibull distribution parameters are computed for the full dataset as well as each subset. Although the convergence of the implemented optimization algorithm is not successful all the time, future studies could research methods to provide better initial guesses or use commercial software packages to estimate the parameters.

To evaluate the goodness-of-fit for each distribution, the mean square error and Akaike Information criteria are used to determine the deviations of the model and level of overfitting to the data. Relative efficiencies are computed between the 1-Weibull and 2-Weibull distributions; the results indicated that for most subsets, the 1-Weibull is a better predictor. However, for a few subsets and the recovery duration full dataset, the 2-Weibull distribution is a better model based on the MSE and AIC criteria. When selected, the 2-Weibull distribution also implies potential bimodal behavior in its dataset. The advantage of employing Weibull distributions for modeling is the ease of implementation into other simulations with just a few parameters.

However, in order to generate predictions of safing events based on many user-defined inputs, a new approach for predictions is required. A Gaussian process model met this criteria; it is a type of supervised learning algorithm that is trained and tested using the existing safing event dataset. Implementation of this model is done using the GPML toolbox. Training involves the selection of the data's noise parameter, training ratio, covariance function, mean function, likelihood function, and inference method. Based on the mean square errors for the predicted mean and variance for each testing dataset, training parameters and functions are selected that minimized those errors. Other performance metrics and cross-validation techniques could be utilized in the future to select optimal training parameters. The usage of other likelihood functions such as a Gamma or Weibull is possible within the Gaussian Process framework, referenced as generalized linear models, could help tackle the assumptions made on the dataset and noise. Furthermore, it is recommended that a subject matter expert aid in the training process.

To assess mission inoperability, a generalized Monte Carlo simulation is implemented to quantify the likelihood of realizing the worst-case inoperability rates. Using the Next Mars Orbiter mission concept as a case study, a framework is generated for the GP model that can easily 'plug-and-play' into the main simulation. Using the developed single Weibull distribution, 2-Weibull mixture distribution, and the Gaussian Process as the predictive models, the likelihood of inoperability rates, outage times, and number of safing events are compared. From highest to lowest mission operability rates, the GP model predicts the highest, followed by the mixture distribution, and finally the single Weibull distribution model. Recommendations are made and implications analyzed for the NeMO concept for the predicted likelihood of inoperability rates. This includes increasing spacecraft margins for missed-thrust periods as well as increases in operational and on-board fault management capabilities.

In the area of predictive analytics, this paper uses standard statistical methodologies to understand trends in the dataset, and develops, trains, and tests predictive models for a sample mission scenario - the Next Mars Orbiter. This work is a step towards creating a more complete tool for safing event analysis and prediction using a historical database of past interplanetary spacecraft missions. The results from this paper help influence mission designers to factor in the effects of safing events on spacecraft margins and requirements in order to make design and operational decisions.

ACKNOWLEDGMENTS

The authors would like to thank Rob Lock and the Mars Formulation Office at the Jet Propulsion Laboratory for funding this research and Dr. David Spencer (Purdue) for enabling the funding mechanism to the Georgia Institute of Technology. The authors would like to thank Travis Imken (JPL) for the extensive support, guidance, and feedback during the course of this research. The first author would like to thank Dr. Joseph Saleh (GT) for initial guidance on analyzing reliability data for spacecraft. The first author would also like to thank Marcus Pereira, a graduate student in the Autonomous Control and Decision Systems Lab under Dr. Evangelos Theodorou at Georgia Tech. His guidance on supervised learning algorithms for regressions helped formulate and train the predictive model.

This work was carried out by the Space Systems Design Laboratory at the Georgia Institute of Technology, under contract to the Jet Propulsion Laboratory, California Institute of Technology.

REFERENCES

[1] J. Wertz, D. Everett, and J. Puschell, *Space Mission Engineering: The New SMAD*, ser. Space technology library. Microcosm Press, 2011. [Online]. Available: https://books.google.com/books?id=VmQmtwAACAAJ

[2] *Emerging Capabilities for the Next Mars Orbiter*. Mars Exploration Program Analysis Group (MEPAG), February 2015.

[3] T. K. Imken, T. M. Randolph, M. DiNicola, and A. K. Nicholas, "Modeling spacecraft safe mode events," *IEEE Aerospace Conference*, March 2018.

[4] J.-F. Castet and J. H. Saleh, "Satellite reliability: Statistical data analysis and modeling," *Journal of Spacecraft and Rockets*, vol. 46, no. 5, pp. 1065–1076, Sep 2009. [Online]. Available: https://doi.org/10.2514/1.42243

[5] G. F. Dubos, J.-F. Castet, and J. H. Saleh, "Statistical reliability analysis of satellites by mass category: Does

spacecraft size matter?" *Acta Astronautica*, vol. 67, no. 5-6, pp. 584–595, 2010.

[6] E. Siegel, *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die*. Wiley, 2016.

[7] C. Nyce, "Predictive analytics white paper," American Institute for CPCU, Insurance Institute of America, Tech. Rep., 2013.

[8] J. H. Saleh and J.-F. Castet, *Spacecraft reliability and multi-state failures: a statistical approach*. John Wiley & Sons, 2011.

[9] D. Montgomery and G. Runger, *Applied Statistics and Probability for Engineers, Binder Ready Version*. Wiley, 2013. [Online]. Available: https://books.google.com/books?id=nflRnQEACAAJ

[10] E. Feigelson and G. J. Babu, "Beware the kolmogorov-smirnov test!" [Online]. Available: https://asaip.psu.edu/Articles/beware-the-kolmogorov-smirnov-test

[11] E. E. Elmahdy and A. W. Aboutahoun, "A new approach for parameter estimation of finite weibull mixture distributions for reliability modeling," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 1800 – 1810, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0307904X12002545

[12] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, Dec 1974.

[13] C. Bishop, *Pattern Recognition and Machine Learning: All "just the Facts 101" Material*, ser. Information science and statistics. Springer, 2013. [Online]. Available: https://books.google.com/books?id=HL4HrgEACAAJ

[14] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptative computation and machine learning series. University Press Group Limited, 2006. [Online]. Available: https://books.google.com/books?id=vWtwQgAACAAJ

[15] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *Journal of Machine Learning Research*, vol. 11, no. Nov, pp. 3011–3015, 2010.

[16] C. Rasmussen and C. Williams, *The GPML Toolbox version 4.1*, November 2017, http://www.gaussianprocess.org/gpml/code/matlab/doc/manual.pdf.

BIOGRAPHY

Swapnil R. Pujari *received a B.S. and M.S. in Aerospace Engineering from the Georgia Institute of Technology in 2016 and 2018 respectively. At his time at Georgia Tech, Swapnil was involved in small satellite missions. He was the mechanical lead and Chief Systems Engineer of the Prox-1 Microsatellite mission leading and personally conducting much of the design, integration, and testing of the physical spacecraft. Additionally he was the payload lead to the Tether and Ranging (TARGIT) CubeSat Mission and the electrical power subsystem lead on MicroNimbus, a 3U radiometer CubeSat Mission. He has interned twice with the Jet Propulsion Laboratory under the Mars Formulation Office. There he worked on the Next Mars Orbiter and Mars Sample Return Lander concept missions working with CAD, orbit design, system budget analyses, day-in-the-life simulations, and telecommunications analyses.*

E. Glenn Lightsey *is a Professor in the Daniel Guggenheim School of Aerospace Engineering at the Georgia Institute of Technology. He received his Ph.D. from Stanford University in 1997. He is the Director of the Space Systems Design Lab at Georgia Tech. His research program focuses on the technology of satellites, including: guidance, navigation, and control systems; attitude determination and control; formation flying, satellite swarms, and satellite networks; cooperative control; proximity operations and unmanned spacecraft rendezvous; space based Global Positioning System receivers; radio navigation; visual navigation; propulsion; satellite operations; and space systems engineering. He has written more than 130 technical publications. He is an AIAA Fellow, and he serves as Associate Editor-in-Chief of the Journal of Small Satellites and Associate Editor of the AIAA Journal of Spacecraft and Rockets.*

APPENDIX

Table 6 shows all of the missions that are contained in the safing event database. It lists how the missions are classified for the mission class, destination, duration, and solar electric propulsion categories.

Tables 7 and 8 show the computed p-values from the Chi-Squared hypothesis test between each mission classifier for the time-between-events and recovery duration datasets.

Tables are presented on the next page.

Table 6. Mission Classifier Inputs

Mission Name	Class	Destination	Duration [years]	Electric Propulsion
Dawn	Discovery	Asteroid/Comet	10-15	Yes
Deep Impact	Discovery	Asteroid/Comet	5-10	Yes
Deep Space 1	Discovery	Asteroid/Comet	0-5	No
Genesis	Discovery	Heliophysics/Exoplanet	0-5	No
Lunar Reconnaissance Orbiter	Discovery	Moon	5-10	No
Mars Atmosphere and Volatile Evolution	Discovery	Mars	0-5	No
Mars Climate Orbiter [2]	Discovery	Mars	0-5	No
Mars Global Surveyor	Discovery	Mars	5-10	No
Mars Odyssey	Discovery	Mars	15-20	No
Mars Polar Lander [2]	Discovery	Mars	0-5	No
Phoenix Mars Lander [2]	Discovery	Mars	0-5	No
Stardust	Discovery	Asteroid/Comet	10-15	No
Juno	New Frontiers	Jupiter	5-10	No
Kepler	New Frontiers	Heliophysics/Exoplanet	5-10	No
Mars Reconnaissance Orbiter	New Frontiers	Mars	10-15	No
New Horizons	New Frontiers	Kuiper Belt Object	10-15	No
OSIRIS-REx	New Frontiers	Asteroid/Comet	0-5	No
Sptizer	New Frontiers	Heliophysics/Exoplanet	5-10	No
Cassini	Flagship	Saturn	15-20	No
Galileo	Flagship	Jupiter	10-15	No
Mars Science Laboratory [2]	Flagship	Mars	0-5	No

Table 7. Contingency Table: Chi-Squares p-Values for Time-Between-Events

	Mission Class	Mission Duration	Mission Destination	Electric Propulsion	Safing Event Cause	Safing Event Mission Phase
Class	-	1.61E-05	4.89E-46	3.54E-05	4.02E-04	2.57E-04
Duration	1.61E-05	-	6.76E-31	0.010851	1.40E-03	9.08E-05
Destination	4.89E-46	6.76E-31	-	3.6702E-11	6.03E-05	3.26E-19
SEP	3.54E-05	0.010851	3.67E-11	-	0.409	0.374
Cause	4.02E-04	1.40E-03	6.03E-05	0.409	-	0.0957
Phase	2.57E-04	9.08E-05	3.26E-19	0.374	0.0957	-

Table 8. Contingency Table: Chi-Squares p-Values for Recovery Duration

	Mission Class	Mission Duration	Mission Destination	Electric Propulsion	Safing Event Cause	Safing Event Mission Phase
Class	-	1.78E-06	9.85E-29	9.80E-05	0.420	0.0484
Duration	1.78E-06	-	2.76E-32	1.39E-4	0.114	9.75E-05
Destination	9.85E-29	2.76E-32	-	8.37E-12	0.0209	2.71E-10
SEP	9.80E-05	1.39E-4	8.37E-12	-	0.484	0.649
Cause	0.420	0.114	0.0209	0.484	-	0.00248
Phase	0.0484	9.75E-05	2.71E-10	0.649	0.00248	-