BAYESIAN INTERVAL ESTIMATION AND PREDICTIVE ANALYSIS IN A NONHOMOGENEOUS POISSON PROCESS WITH DELAYED S-SHAPED INTENSITY FUNCTION

OTIENO COLLINS

A Thesis Submitted to the Graduate School in Partial Fulfilment of the Requirements for the Master of Science Degree in Statistics of Egerton University

EGERTON UNIVERSITY

MARCH 2024

DECLARATION AND RECOMMENDATION

Declaration

This thesis is my original work and has not been presented in this university or any other for the award of a degree.

Signature: …… ……… Date: …01/03/2024……..............

Otieno Collins

SM123/14596/18

Recommendation

This thesis has been submitted with our approval as university supervisors.

Signature: … …. … Date: …07/03/2024………………

Dr. Luke Akong'o Orawo (Ph.D.)

Egerton University

Signature: … ...…… Date: ……08/03/2024………….

Dr. George Munene Matiri (Ph.D.)

Egerton University

DEDICATION

I dedicate this work to my dear parents, Mr. Dickson Omenda Onyango & Mrs. Josephine Auma Omenda, and the entire Omenda's family, all of whom I attribute my success to their unconditional support.

ACKNOWLEDGEMENTS

First, I thank the Almighty God for granting me good health, wisdom, patience, and knowledge throughout this research. Glory and Honour be to the Most High God, for His love is unfailing and endless, and His grace is sufficient for all of us. Amen. I also thank the entire staff of Egerton University and, more so, the Mathematics department staff for creating a friendly learning environment. I most sincerely acknowledge the invaluable support from my supervisors, Dr. Luke Akong'o Orawo and Dr. George M. Matiri. Their selfless support, positive criticism, and guidance throughout the research were pivotal. Dr. Luke Akong'o Orawo was very resourceful, and his knowledge, experience, and programming skills were helpful throughout the research. Thank you so much, "Prof," and may God reward you according to His riches and will. I also thank Dr. Obwoge for his support throughout the study: his programming techniques were crucial.

With pleasure, I want to thank my dear parents, Dickson Omenda Onyango and Josephine Auma Omenda, for their prayers and financial support throughout my life and more so throughout this study. Their tireless efforts and hard work encouraged me a lot. May the Lord bless you abundantly and reward you for being caring and responsible parents.

I also want to appreciate my colleagues, Flavian Awere, Samwel Karanja Kamau, Samson Tsuma, Francis Kariuki, Erick Ogola, Wesley Ochoka, and Jackline Akomboh, for their support in my academic and social life. I also humbly express my gratitude to all my siblings, more so Janet Akinyi and Okoth Joseph, for their generous support while I was pursuing my course. To all my friends and to the many people who are not mentioned but contributed to the completion of this work in any way, I hereby recognize your support. May God bless you all. Amen.

ABSTRACT

In the 21st century, software reliability is a significant issue as computers are the most preferred system in almost every global sector. A software is reliable if it can perform its functions for a specified period under specified conditions without causing system failure. A software neither wears out nor burns out and does not fail unless flaws within cause a failure in its dependent system. As such, software reliability testing is performed in the development phase to correct the flaws within the software. Among the non-homogeneous Poisson processes (NHPP) software reliability growth models (SRGMs) proposed and used in software reliability assessment is the Delayed S-shaped model with two unknown parameters α and β, that must be estimated. Most research works have fitted the model to software failure data and obtained point and interval estimates of the unknown parameters using the Maximum Likelihood Estimation (MLE) and Bayesian approaches. However, the construction of Bayesian credible sets for the parameters of this model and the comparison of their accuracy with the traditional Wald confidence intervals based on simulation has not been explored. Predictive analysis on the model has been explored using the Bayesian method with gamma-distributed informative prior. More optimal methods can be developed based on the priors assigned to the unknown parameters to enhance accuracy in modifying, debugging, and determining when to terminate software testing processes. This study introduced a non-informative prior given by $1/\alpha\beta$ and also used $1/\alpha$ prior existing in the literature and gamma-distributed informative prior to construct Bayesian credible intervals, compare them with Wald confidence intervals using interval lengths and coverage probabilities, and perform predictive analysis. Markov Chain Monte Carlo (MCMC) via Metropolis-Hastings (MH) within Gibbs was used to sample the parameters from their respective conditional posterior distributions. Bayesian approach was also used to address four prediction issues closely associated with software reliability testing. The issues have been outlined as Propositions I, II, III, and IV for the case of non-informative priors, and I.1, II.1, III.1, and IV.1 for the case of the informative prior. The study found that the Bayesian method with gamma-distributed informative and $1/\alpha\beta$ priors yielded more precise interval estimates than the Wald confidence intervals. Moreover, the study developed methods for addressing the outlined single-sample prediction problems and illustrated them using secondary software failure data. The methods developed in this study can be used in software quality assessment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

NHPP	Non-Homogeneous Poisson Process
SRGM	Software Reliability Growth Model
SRM	Software Reliability Modelling
GO	Goel-Okumoto
HPP	Homogeneous Poisson Process
AT&T	American Telecommunication and Telegraph
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
MLE	Maximum Likelihood Estimation
MLP	Maximum Likelihood Principle
UPL	Upper Prediction Limit
HPD	Highest Posterior Density
PLP	Power Law Process
GO	Goel-Okumoto
AE	Accuracy of Estimate
MSE	Mean Squared Error
pdf	Probability Density Function

LIST OF SYMBOLS

α	Scale parameter for the Delayed S-shaped software reliability model
β	Shape parameter for the Delayed S-shaped software reliability model
$\hat{\alpha}$	Estimator of the scale parameter
$\hat{\beta}$	Estimator of the shape parameter
$N(t)$	Cumulative number of software failures observed after time t
t_i	i^{th} failure time
$\lambda(t)$	Failure intensity function at time t
m (t)	the mean value function
$\pi(\alpha, \beta)$	Joint prior distribution for α and β
$\pi(\alpha)$	Prior distribution for α
$\pi(\alpha, \beta \mid \underline{t})$	Bayesian posterior density
n	the total number of software failures
C_p	Coverage probability
$f\left(t^{+} \middle| \underline{t}\right)$	Bayesian posterior predictive distribution of t^{+}

CHAPTER ONE

INTRODUCTION

1.1 Background Information

Daily life situations yield random processes identified as Poisson processes. Some of these processes depend on time and are modelled as Non-Homogeneous Poisson Processes (NHPP). An NHPP is a mathematical model describing the occurrence of events over time, where the average rate of occurrence varies with time and is represented by an intensity function. Practically, most real-world processes exhibit variations in their average rate of occurrence over time, making NHPP models widely used in several applied fields, such as communication, transport, businesses, imaging, and genomics. A good example is software development, where researchers have proposed different NHPP models for software quality testing. In the 21st century, software reliability has become a significant issue as computers are the most preferred system in various operations in different sectors, including healthcare, aircraft, industrial and quality control processes, and surveillance. Software is reliable if it can accomplish its defined functions for a specified period of time under a given environment (Pham & Pham, 2000). Thus, software reliability is the probability of software operating for a specified period under specific conditions without causing system failure (Lai & Garg, 2012; Robles, 2011).

In the software development phase, some errors may be present in the software or could be introduced in the development process. The errors have to be detected and removed to ensure reliability and prevent failure. An error is a cause of failure, more precisely defined as an improper deviation from a normal procedure (Yamada *et al.*, 1984). While it is possible to have fault-free software that will not fail, it can only be achieved if the faults in the code have been removed. One important feature of the software is that it neither wears out nor burns out and does not fail unless flaws within cause a failure in its dependent system (Lai & Garg, 2012). Failure-free software presents numerous benefits to software developers, including increased software quality, improved testing productivity, reduced testing costs, and improved release time to market (Lai & Garg, 2012).

Software reliability modelling comes in handy when addressing software quality issues. Software reliability assessment has used the NHPP software reliability growth models (SRGMs), yielding reliable results. SRGMs are mathematical functions that illustrate how software errors are detected and removed, precisely defined as technologies that enable the quantitative assessment of software reliability (Inoue & Yamada, 2006). The NHPP SRGMs have accurately captured the error detection and removal processes, enabling the estimation of costs for maintaining undiscovered faults. Generally, this process of using various models to assess software reliability is known as software reliability modelling (SRM). SRM aims to describe the fault-related behaviour of software testing processes, including error detection, introduction, and removal (Xie *et al.,* 2007).

Over the past decades, researchers have proposed several NHPP models for software reliability assessment. The NHPP delayed S-shaped SRGM was developed by Yamada *et al.* (1983) by modifying the earlier versions of the Goel-Okumoto (GO) model to introduce a new model with a mean value function that exhibits an S-shaped growth curve. Yamada *et al.* (1984) modified their earlier version to propose a software quality assessment method integrating capture-recapture sampling and SRGM. The model is unique and more appropriate because it assumes a delay between the first failure observation time and the reporting time (Lee *et al.,* 2014). The NHPP Delayed S-shaped model has a mean value function that reflects the delay in failure reporting time. It has two unknown parameters denoted by α and β, where α is the expected total number of errors in software before testing, and β is the failure detection rate per fault (Hanagal & Bhalerao, 2018). Over the years, various researchers have derived point estimates of the parameters of this model and constructed interval estimates using either Maximum Likelihood Estimation (MLE) or the Bayesian approaches. In the MLE, the sampling distributions of the estimators are assumed to be asymptotically normal. However, interval estimates based on this approach may be inaccurate and biased for small sample sizes. Given the limited availability of historical data, particularly in the case of new software projects, software developers often face challenges in accurately estimating reliability (Lee *et al.,* 2014). In such scenarios, the Bayesian approach is reasonable, enabling more informed decisions and mitigating uncertainties associated with insufficient historical data. The Bayesian approach requires predefining prior distributions of the parameters, which could be either informative or non-informative.

The main goal of software quality testing is to identify and rectify errors, defects, and potential issues within the software during development. Predictive analyses have been performed on the NHPP Delayed S-shaped model using the Bayesian approach. Predictive analysis is crucial in determining the optimal time to terminate the software development testing phenomenon (Yu *et al.,* 2007). Both estimation and prediction may be based on the number of faults or inter-failure times. The time required to correct a software error cannot be neglected in the testing process: each detected error has to be reported, diagnosed, removed, and verified before concluding that it is corrected (Xie *et al.,* 2007). Moreover, the accuracy in estimation and prediction depends on the methods used in constructing posterior distributions, such as prior functions. Until now, a flat prior given by $1/\alpha$ and gamma-distributed informative prior have been used with the Delayed S-shaped model by Yin and Trivedi (1996) and Lee *et al.* (2014), respectively. However, gamma-distributed informative prior has been used in predictive analysis but not interval estimation. Depending on the prior distributions assigned to the unknown parameters of the model, more optimal methods can be developed based on the Bayesian approach. Moreover, more issues closely related to software development testing can be addressed using this model. Therefore, this study constructed Bayesian credible intervals, compared them with Wald interval estimates, and conducted Bayesian predictive analysis using a gamma-distributed informative prior and two non-informative priors, $1/\alpha$ and $1/\alpha\beta$, addressing issues that may be of interest during software quality testing.

1.2 Statement of the Problem

The application of computers in various sectors is increasing at a higher rate, leading to major considerations about software reliability. The NHPP Delayed S-shaped SRGM has been used in several reliability testing processes and has yielded reliable results. Most research works have fitted the Delayed S-shaped model to software failure data, in which point and interval estimates of its parameters have been obtained using either the MLE or the Bayesian method. However, the construction of Bayesian credible sets for the parameters of this model and the comparison of their accuracy with the traditional Wald confidence intervals based on simulation has not been explored. Given the challenge of limited historical data in software quality testing, simulation provides a more efficient approach for evaluating model performance. Moreover, the rapid changes in software development require optimal methods to improve reliability testing processes by enhancing accuracy in estimation and prediction. The Bayesian approach is advantageous as it can yield reliable results even when the sample sizes are small since it incorporates prior information. Existing literature has considered the joint prior distribution given by $1/\alpha$, while the gamma-distributed informative prior has not been used in interval estimation. This study introduced a non-informative prior given by $1/\alpha\beta$ and extensively used gamma-distributed informative prior to construct Bayesian credible intervals and perform predictive analysis. The study compared Bayesian credible intervals with Wald confidence intervals using coverage probabilities and interval lengths. Predictive analysis was performed to address software testing issues closely related to software development, which had not been addressed using the Delayed S-shaped model.

1.3 Objectives

1.3.1 General Objective

To carry out a study on Bayesian interval estimation and predictive analysis on the Delayed S-shaped software reliability model with intensity function.

1.3.2 Specific Objectives

i. To obtain Bayesian credible intervals for the shape and scale parameters of the NHPP Delayed S-shaped SRGM using informative and non-informative priors.
ii. To compare Bayesian interval estimates and Wald confidence intervals using coverage probabilities and interval lengths.
iii. To perform predictive analysis on the model using informative and non-informative priors.

1.4 Assumptions of the Study

i. For the software failure times given by $0 < t_1 < t_2 < \cdots < t_n < T$, if T is time truncated, then T = t_n.
ii. Simulation emulates software end-user environment and can generate inter-failure times data for the study.

1.5 Justification

Software developers are often interested in the reliability of their software before introducing it into the market. Software end-users also need assurance that the software they acquire from the market will perform maximally throughout the operational lifespan, yielding expected output. Due to costs associated with software development, software warranties are hardly provided, and producers often attempt to avoid any responsibility for software failure once the end-user has acquired it. Moreover, the software development process faces inevitable errors arising during designing and testing. The errors may affect systems' durability and performance if not removed. To address such concerns, a software reliability test is conducted during the development phase, ensuring that the testing cost is affordable and the test results are reliable and appropriate for decision-making. Software development faces rapid changes that require developing optimal methods to increase accuracy in testing and prediction. The improved accuracy will ensure that software developers incur low costs in production but still produce high-quality software. Such desired results can be achieved through estimation and predictive analysis of the NHPP Delayed S-shaped software reliability model. The model assumes a delay between the first failure observation time and reporting time, which reflects the real testing scenario. Simulation provides a more efficient method for estimating model parameters and predicting future failure occurrences. The developed methods will contribute to accuracy in estimation and prediction, yielding results that can be used in software quality assessment.

CHAPTER TWO

LITERATURE REVIEW

This section provides a review of other studies focusing on software reliability and the application of NHPP Delayed S-shaped SRGM in software quality testing.

2.1 Counting Process

A counting process $N(t)$ is a stochastic process with positive integers and non-decreasing values, that is, $N(t) \geq 0$ for $t > 0$. The process is concerned with the number of occurrences of an event within a specified period of time. The events are assumed to occur randomly with the process, holding that if $r < t$, then $N(t) - N(r)$ is the number of occurrences of an event within the interval (r, t). Counting processes are extensively used in software reliability analysis since the errors that lead to a software failure occur at random. In this regard, error detection also occurs randomly. Homogeneous Poisson processes (HPP) and NHPP are two examples of commonly used counting process probabilistic models.

2.1.1 Poisson Process

The Poisson process is a model for countable events that randomly occur at a certain rate, with a known average time between them. Such events form Poisson random variables with parameter μ and a PMF given by:

$$P_x(t) = \frac{e^{-\mu} \mu^t}{t!} \tag{2.1}$$

A counting process $N(t)$ is said to be a Poisson process if the following conditions are satisfied:

(i) $N(0) = 0$: the failure at time zero.

(ii) Given any points t such that $t = t_0 < t_1 < t_2 \ldots < t_n$, the random variables $N(t_0, t_1]$, $N(t_1, t_2]$, … $N(t_{n-1}, t_n]$ are independent random variables, which is the independent increment property.

(iii) There exists a function λ such that $\lambda(t) = \lim_{\Delta(t)\to 0} Pr \frac{\{N(t+\Delta t) - N(t) \geq 1\}}{\Delta(t)}$

(iv) $\lim_{\Delta(t)\to 0} \Pr \frac{\{N(t+\Delta t) - N(t) \geq 2\}}{\Delta(t)} = 0$

These properties imply that:

$$\Pr(N(t) = n) = \frac{1}{n!}\left(\int_0^t \lambda(x)\, dx\right)^n \exp\left(-\int_0^t \lambda(x) dx\right) \tag{2.2}$$

2.1.2 Homogenous Poisson Process

An HPP is a counting process that describes events occurring randomly within a given time. A counting process $\{N(t), t \geq 0\}$ is an HPP if it satisfies the following conditions.

i. The failure at time zero, $N(0) = 0$.

ii. The intensity function, $\lambda(t)$, is constant.

iii. The number of events occurring at any interval of length $t = t_2 - t_1$ has a Poisson distribution with the mean λt.

$$P[N(t_2) - N(t_1) = n] = \frac{e^{-\lambda t}(\lambda t)^n}{n!}, 0 \leq t_1 \leq t_2, \; n = 0, 1, \dots. \quad (2.3)$$

iv. The process has independent increments and stationary increments. That is, a point process has stationary increments if, for all k, $[P\,r(n, n+s) = k]$ is independent of t.

It further has the following assumptions:

i. Two or more events cannot occur at the same period in time.

ii. Events are independently and randomly generated.

An HPP has the following properties:

i. A process is an HPP with constant intensity function λ if and only if the times between events are independent and identically distributed exponential random variables with mean $\frac{1}{\lambda}$.

ii. If $0 < t_1 < t_2 < \cdots < t_n$ are the failure times from an HPP, then the joint probability density function (pdf) of $t_1, t_2, \dots, t_n$ is given by;

$f(t_1, t_2, \dots t_n) = \lambda^n e^{-\lambda t_n}, 0 < t_1 < t_2 < \cdots < t_n$

iii. For a system modelled by an HPP, the waiting time to the n^{th} failure has a gamma distribution with parameter $\alpha = n, \beta = 1/\lambda$

iv. Conditional on $N(t) = n$, $0 < t_1 < t_2 < \cdots < t_n$ are distributed as order statistics from a uniform distribution in the interval $(0, T)$.

v. The probability of failure occurrence after time T is $Pr[t > T] = Pr[N(T) = 0] = e^{-\lambda T}$

2.1.3 Nonhomogeneous Poisson Process

An NHPP represents the number of failures that occur in a time interval $t_0 < t_1 < t_2 < \cdots < t_n$ up to time $t_n = t$. It describes random processes with an average rate of occurrence that varies over time. NHPPs are used in software reliability testing to determine a convenient mean value function, indicating the expected number of failures up to a specified time *t*. Therefore, a counting process is said to be an NHPP if it satisfies the following conditions:

i. The initial condition is $N(0) = 0$

ii. The process has independent increment property such that $N(t_0, t_1), N(t_1, t_2), \ldots$, $N(t_{n-1}, t_n)$ are independent random variables for any specified time points, $t_0 < t_1 < t_2 < \cdots < t_n$, where $t_0 = 0$.

iii. The probability that *n* failures occur at a time interval of (0, t) is given by $Pr\,\{N(t) = n\} = \frac{[m(t)]^n}{n!} e^{-m(t)}$, where $m(t)$ is the mean value function denoting the expected number of failures and is given by $m(t) = \int_0^t \lambda(t)dt$, and $\lambda(t)$ is the intensity function.

iv. The failure rate of the process is given by the probability that exactly one failure occurs within a small-time interval Δt denoted by:
$Pr(t, t + \Delta t) = Pr\,\{N(t + \Delta t) - N(t) = 1\} = \lambda(t)\,\Delta t + o(\Delta t)$

v. The probability that more than one failure occurs within a small interval of time, Δt is negligible, that is, $Pr\,\{N(t + \Delta t) - N(t) \geq 2\} = o(\Delta t)$.

An NHPP has the following properties:

i. The joint pdf of the failure times $t_1, t_2, \ldots, t_n$ from a NHPP with intensity function λ(t) is given by:
$f(t_1, t_2, \ldots, t_n) = (\prod_{i=1}^{n} \lambda(t_i)) \exp\left(- \int_0^T \lambda(t)dt\right)$
where T is the stopping time: $T = t_n$ for the failure truncated case, and $T = t$ for the time truncated case.

ii. If $0 < t_1 < t_2 < \cdots$ are the epochs at which the failure times occur, then the inter-occurrence intervals $T_k = t_k - t_{k-1},\ k = 1, 2, \ldots$ are independent random variables with densities:
$f_{t_k}(t_k) = \lambda(t_k) exp\left(- \int_{t_{k-1}}^{t_k} \lambda(t)dt\right)$
Furthermore, $\{t_1, t_2, \ldots\}$ is a Markov sequence with transition density given by $Pr(t_k | t_{k-1}) = \lambda(t_k) \exp\left(- \int_{t_{k-1}}^{t_k} \lambda(t)dt\right)$

iii. Conditional on $N(t) = n$, the epochs at which failures occur, $0 < t_1 < t_2 < \cdots < t_n \leq T$ have the same distribution as n order statistics corresponding to a random sample of n observations from the density:
$f(t) = \lambda(t) / \int_0^T \lambda\,(t)dt\,, 0 \leq t \leq T,$

which reduces to the uniform distribution over $[0, T]$ when $\lambda(t) = \lambda$.

2.2 Nonhomogeneous Poisson Process Software Reliability Models

NHPP models have been extensively used in software reliability testing to capture fault-related behaviours, including detection, isolation, and removal. The NHPP models are commonly used to assess the reliability growth of repairable software systems and forecast failure occurrences (Yu *et al.,* 2007). According to the repair policy, when a failure occurs, systems can be categorized as non-repairable and repairable systems (Zhao *et al.*, 2017). In particular, the NHPP can describe the failure-repair process under minimal repairs. The minimal repair action means that the system is restored to functioning but is only as good as other equipment equal to its age at failure (Zhao *et al.,* 2017). Software developers aim to build more reliable and quality software. Therefore, they have always needed an appropriate model that would yield desirable results and have utilized the NHPP models due to their reliability and convenience in monitoring the reliability growth of the developed software.

Different authors have developed various NHPP models for reliability assessment. The NHPP models applicable in software reliability tests have accurately estimated and predicted software reliability (Ismael Al Turk, 2019). Examples of NHPP SRGMs include the extended Weibull process with mean value function $m(t) = \frac{a}{b} t^b$ and an intensity function $\lambda(t) = at^{b-1}$ (Yoo, 2018); the Cox and Lewis (1966) model with intensity function $\lambda(t) = e^{\alpha+\beta t}$ (Cid & Achcar, 1999), the Goel-Okumoto (GO) (1979) model with intensity function $\lambda(t) = \alpha\beta e^{-\beta t}$ (Goel & Okumoto, 1979); the Musa-Okumoto (1984) model with $\lambda(t) = \frac{\alpha}{(t+\beta)}$ (Musa & Okumoto, 1984), and the Delayed S-shaped model with $\lambda(t) = \alpha\beta^2 t e^{-\beta t}$ (Yamada *et al.,* 1983).

The models can be classified as perfect and imperfect debugging NHPP models. The perfect debugging models assume that errors are removed with certainty without introducing further errors into the software. They include the Yamada Delayed S-shaped model, the GO model, the Generalized Goel NHPP model, the Gompertz growth curve model, the Inflection S-shaped model, the Logistic growth curve model, the Modified Duane Model, and the Musa-Okumoto model (Kaur & Panwar, 2015). On the other hand, imperfect debugging models capture a scenario where the detected faults are not removed with certainty or new faults are introduced during error correction. They include the Pham-Nordmann-Zhang (PNZ) model, Pham-Zhang (P-Z) model, Yamada Rayleigh, Yamada Imperfect Exponential model, and Yamada Exponential model (Kaur & Panwar, 2015).

The NHPP SRGMs have been developed based on two primary assumptions: the error-detection rate per error is constant during testing, or it depends on the elapsed time of testing. The first assumption implies the errors are independent and equally likely to be detected. The second assumption further validates another assumption that software testing detects two classes of errors: errors that are easy to detect and those that are difficult to detect in the testing process (Kimura *et al.*, 1992). The former class of errors can be detected early, while the latter class can be detected later in the testing process, leading to a different curve of the cumulative number of detected errors. Such models show that the growth curve of the cumulative number of detected errors forms an exponential curve in the initial stage of the testing phase and is later followed by an S-shaped curve and an exponential curve, respectively (Kimura *et al.,* 1992). An example of the model developed under the second assumption is the one presented by Yamada *et al.* (1985), with a mean value function given by:

$m_p(t) = \sum_{i=1}^{2} p_i a(1 - e^{-b_i t})$

The model can be generalized for the case where the difficulty of error detection is of some degrees such that it is incorporated by assuming k types of errors. The resulting NHPP model has a mean value function given by Yamada *et al.* (1985).

$m_p(t) = \sum_{i=1}^{k} p_i a(1 - e^{-b_i t})$, where $\sum_{i=1}^{k} p_i = 1$, $p_i > 0$, and $0 < b_k < b_{k-1} < \cdots < b_2 < b_1$.

2.3 Bayesian Method

Bayesian estimation involves using the likelihood function and prior density function to derive a posterior distribution by means of the Bayes Theorem (Chulani *et al.*, 1999). Let $X_1, \cdots, X_n$ be independent observations, each distributed according to a density function $f(x/\theta)$, depending on a parameter θ taking values in parameter space Θ. Let $\pi(\theta)$ denote the prior density on Θ. The posterior distribution function of θ is given by:

$$\pi(\theta/\underline{x}) = \frac{f(\underline{x}/\theta)\pi(\theta)}{m(\underline{x})}, \qquad (2.4)$$

where $f(\underline{x}/\theta)$ is the likelihood function, and $m(\underline{x}) = \int f(\underline{x}/\theta)\pi(\theta)d\theta$ is the marginal distribution of the observations. The Bayes interval and point estimates of the parameters of interest are then generated from the posterior distribution. The mean, median, and mode of the posterior distribution have been used as possible Bayes estimates of the parameter of interest θ. The concept of the Bayesian credible interval is defined as follows. Suppose that ψ is some fixed number in the interval (0,1). A subset Θ^* of the parameter space Θ is called a

$100(1-\psi)\%$ credible set for the unknown θ if and only if the posterior probability of the subset $\Theta^*$is at least $1-\psi$. When the parameter space Θ is a subset of the real line, then Θ^* is an interval and is naturally referred to as a $100(1-\psi)\%$ credible interval. The narrowest credible set, which includes only those values of θ with the highest posterior, can be chosen from a long list of competing credible sets. Such a credible set is called the highest posterior density (HPD). For some fixed $\alpha \in (0,1)$, an HPD credible region Θ^*has the following form:

$$\Theta^* = \{\theta \in \Theta: \pi(\theta/\underline{x}) > a\}, \quad (2.5)$$

where $a = a(\psi, \underline{x})$ is the largest constant such that the posterior probability of Θ^* is at least $1-\psi$.

Suppose that the posterior density $\pi(\theta/\underline{x})$ is unimodal and is symmetric about its finite mean (which would coincide with its median), then $100(1-\psi)\%$ HPD credible interval Θ^*will be the shortest and equal-tailed and symmetric about the posterior mean.

Another goal of Bayesian inference of interest is the prediction of future observations. Making reliable predictions based on the observed data is one of the primary tasks in probability and statistics. The Bayesian approach is one of the predictive techniques that have been explored, especially in software reliability prediction. The approach uses the posterior distribution, $\pi(\theta/y)$, to make inferences about the parameter θ. The distribution of unknown but observable data, y, also known as the prior predictive distribution, is given by:

$$\begin{aligned} f(y) &= \int \pi(\theta/y)d\theta \\ &= \int \pi(\theta)\pi(y/\theta)d\theta \end{aligned} \quad (2.6)$$

Bayesian predictive analyses utilize posterior distribution, $\pi(\theta/\underline{x})$, of the model parameter of interest θ to determine the posterior predictive distribution:

$$f(x_{n+k}/\underline{x}) = \int f(x_{n+k}/\theta)\pi(\theta/\underline{x})d\theta, \quad (2.7)$$

which describes the beliefs about the future observation x_{n+k} given the observed data $\underline{x}$. As Rahman *et al.* (2016) noted, predictive inference is based on observable data, with Bayesian statistical analysis focusing on posterior distribution for the parameters as the main objective. The posterior predictive distribution is used to make inferences about the future unknown observation, that is, predicting a possible value of a future observation and constructing a prediction interval in which the future observation would lie with some probability.

The increasing use of the Bayesian method in different fields of study is attributed to its numerous advantages relating to flexibility and accuracy in estimation and prediction. The

Bayesian approach is advantageous since it allows the combination of prior information with more recent information obtained from field data or tests (Cheruiyot *et al.,* 2019). Bayesian statistical methods can be applied to any data to yield rich information about the parameters of a model, regardless of the model that motivated the collection of such data (Kruschke, 2010). This flexibility has been of significance in software reliability assessment, where insufficient historical data is an issue. When testing the reliability of newly developed software, developers may have insufficient historical data, making Bayesian analysis reasonable because it incorporates experts' opinions to enhance decision-making (Lee *et al.*, 2014). Moreover, the number of software failures observed during testing is limited, making software reliability estimation involve the risk of large estimation errors (Hirata *et al.,* 2009). As such, the Maximum Likelihood approach may not yield accurate software reliability estimates since its estimation errors can only be reduced by increasing the number of observations. In this regard, Bayesian method has become more prevalent in software reliability testing.

The Bayesian method has been used to estimate and predict software reliability using different software reliability models based on software failure data. Yu *et al*. (2007) and Tian *et al.* (2011) considered the application of the Bayesian method in the NHPP with a power intensity law, with the latter assuming a case of left truncated data. Left truncation may result from the failure to observe in the early stage of the development phase of the testing process or if the importance of a reliability growth model is recognized after manufacturers have severally reported software failures (Tian *et al.,* 2011). The research was based on the Power-Law Process (PLP) or Weibull process and the application of the Bayesian method to parameter estimation and prediction based on the model. They addressed issues in single-sample and two-sample prediction using informative and non-informative priors. Yin and Trivedi (1999) also used the Bayesian method to estimate credible intervals of NHPP-based software reliability models. They considered the GO and Delayed S-shaped NHPP models, addressing two issues: the lack of or use of simple approximations in computing confidence intervals and the lack of confidence interval analysis in software reliability prediction. Nevertheless, they used a non-informative prior given by $\frac{1}{\alpha}$, and provided a numerical example using real data.

In other studies, Lee *et al.* (2014) used the Bayesian method in software reliability assessment based on the Yamada Delayed S-shaped model. They performed Bayesian predictive analysis using joint gamma-distributed informative prior for the parameters and developed an optimal cost model, assuming the model parameters are mutually independent. Akuno *et al.* (2014a) and Akuno *et al.* (2014b) research were based on the GO model, using the Bayesian approach to address issues in two-sample and single-sample prediction, respectively. Moreover, Cheruiyot *et al.* (2018) and Cheruiyot *et al.* (2019) conducted a study on the Musa-Okumoto model using the Bayesian approach, focusing on four single-sample issues in software reliability, using non-informative and informative priors, respectively. Achcar *et al.* (1998) performed Bayesian inference using different models, including General Order Statistics (GOS), Generalized Gamma Order Statistics, and Log-Normal Order Statistics software reliability models. Wahono (2015) conducted a systematic literature review of software defect prediction to address several issues, including determining the methods often used for software defect prediction. The research revealed that Naïve Bayes is among the most frequently used prediction methods in software reliability assessment.

2.3.1 Prior Distributions

The Bayesian method combines prior distributions of the parameters of a model with the likelihood function to obtain a posteriori distribution. The prior distribution indicates the existing state of knowledge or describes the current uncertainty of the parameters of a model before observing the data (Glickman & Van Dyk, 2007). Researchers often incorporate the available information about the unknown parameters in the analysis to estimate model parameters. The main objective of a Bayesian statistical analysis is to obtain the posterior distribution of the model parameters. In this regard, the posterior distribution is the weighted average between the known information about the model parameters before observing the data and the information conveyed in the observed data (Glickman & Van Dyk, 2007).

The choice of a priori distribution is a significant step in Bayesian statistical analysis. Researchers can use available information from experts or their beliefs to formulate prior distributions of the unknown model parameters. The choice of the priors is often guided by the available information about the parameters of interest or intuition (Lee & Vanpaemel, 2018). In this regard, prior distributions are categorized into informative and non-informative priors. Informative priors express specific information about the parameter of interest. Researchers can use their knowledge about the existing problem based on past data and incorporate experts' opinions to formulate a prior distribution that properly reflects their beliefs about the unknown model parameters (Glickman & Van Dyk, 2007). This approach to choosing prior distributions is reasonable because it would be unscientific not to include prior knowledge about the model parameters existing before observing data. If prior information about the parameters of the model is available, informative priors are the most appropriate approach to introducing that information into the model (Golchi, 2018). Furthermore, informative priors can address computational difficulties and improve inference if used appropriately and creatively.

Although informative priors have been commonly used, theory has outlined methods for obtaining non-informative priors to address the limitation of being guided by intuitive knowledge. Such methods include using flat, reference, and Jeffreys priors. Non-informative priors express researchers' beliefs about a particular parameter of interest before considering available evidence. This approach to choosing prior distributions is also referred to as objective, vague, and diffuse and is an attempt at objectivity because it involves acting as though no prior information about the model parameters exists before observing the data (Glickman & Van Dyk, 2007). For instance, Jeffreys explained a method based on the Fisher Information model to derive non-informative priors. The Fisher information matrix is given by:

$$I^F(\theta) = E_{(x|\theta)}\left\{\frac{\partial}{\partial\theta} logf(x|\theta)\right\}^2 \quad (2.8)$$

where X$\sim f(x|\theta)$, and $f(x|\theta)$ is differentiable in θ for every x.

or,

$$I^F(\theta) = -E_\theta[l''(x|\theta)]$$

where $l(x|\theta)$ is the log-likelihood function.

Moreover, for a model with density $f(x|\theta)$, the reference prior is obtained using the formula (Berger *et al.*, 2009):

$$\pi(\theta) = \lim_{k\to\infty} \frac{f_k(\theta)}{f_k(\theta_0)} \quad (2.9)$$

$$f_k(\theta) = \exp\left\{\int p\left(x^{(k)}/\theta\right) log\left[\pi * \left(\theta/x^{(k)}\right)\right] dx^{(k)}\right\} \quad (2.10)$$

where θ is an unknown parameter, θ_0 is an interior point of the parameter space Θ, x is the complete data vector, $\pi^*\left(\theta/x^{(k)}\right)$ is the posterior distribution corresponding to a fixed arbitrary prior $\pi^*(\theta)$, and $x^{(k)} = \{x_1, \dots, x_k\}$ represent *k* conditionally independent replications of x.

2.4 Wald Confidence Intervals

The MLE method has been widely used to compute point estimates and Wald confidence intervals of the parameters of an SRGM. It relies on the normal approximation assumption and is often computed with large samples. The method uses the maximum likelihood principle (MLP) to compute the estimates of the parameters by maximizing the log-likelihood function. The computation requires constructing the Hessian matrix, which is used to estimate asymptotic variances of the parameter estimates. An inverted Hessian matrix yields

diagonal elements that can be used to compute estimator variance (Dalitz, 2018). Wald confidence intervals are computed using the following procedures:
If $f(x/\theta)$ is a function and $L(\theta)$ is the likelihood function, then:

$$L(\theta) = \prod_{i=1}^{n} f(x_i/\theta) \tag{2.11}$$

and

$$l(\theta) = \sum_{i=1}^{n} \log f(x_i/\theta) \tag{2.12}$$

where $l(\theta)$ is the log-likelihood function, θ is the parameter of interest, and *n* is sample size.

If $l(\theta)$ is differentiable and $\theta = (\theta_1, \ldots, \theta_k)$, the MLP yields k equations that can be used to determine the k parameters of θ:

$$\frac{\partial}{\partial \theta_i} l(\theta) = 0 \text{ for } i = 1, \ldots, k \tag{2.13}$$

Suppose $\widehat{\theta_i}$ is the point estimate of the i^{th} parameter, and $sd(\widehat{\theta_i})$ is its standard deviation, a $100(1-\alpha)\%$ Wald confidence interval is given below.

$$\widehat{\theta_i} \pm Z_{\frac{\alpha}{2}} sd(\widehat{\theta_i}) \tag{2.14}$$

Equation (2.13) yields point estimates, while equation (2.14) produces confidence intervals wherein the true parameter θ falls with some probability.

2.5 Coverage Probability

Coverage probability is used to assess the efficiency of parameter estimation methods. It measures the probability that a true parameter value lies within the confidence interval (Dalitz, 2018). The boundaries of the confidence intervals depend on the observed data and, thus, are random variables (Dalitz, 2018). Thus, coverage probability also varies with confidence intervals as data changes. If θ is the true parameter, $(1-\alpha)$ is the predefined coverage probability with which θ falls in the computed confidence interval $[\theta_l, \theta_u]$. Equation (2.15) defines the coverage probability:

$$P_{cov}(\theta) = P\{\theta \epsilon [\theta_l, \theta_u]\} = 1 - \alpha \tag{2.15}$$

2.6 The NHPP Delayed S-Shaped Model

The Delayed S-shaped software reliability growth model was developed by Yamada *et al.* (1983). The model was developed by modifying the GO model to make it S-shaped (Pham & Zhang, 2003). The NHPP Delayed S-shaped model is one in which the software reliability curves assume S-shaped (Hanagal & Bhalerao, 2018). It uses machine execution time as the unit of fault detection or period of removal (Lai & Garg, 2012). The model is given by the function:

$$f(t|\alpha,\beta) = \left(\alpha\beta^2 te^{-\beta t}\right)e^{-\alpha(1-(1+\beta t_n)exp\,(-\beta t_n)} \quad (2.16)$$

The Delayed S-shaped model has the following assumptions (Hanagal & Bhalerao, 2018):

i. Each time a failure occurs, the error that caused it is simultaneously removed, and no other errors are introduced.
ii. The probability of failure detection at any time is proportional to the current number of faults in a software.
iii. All the faults in a software are mutually independent from the failure detection point of view.
iv. Errors in a software lead to system failures at random times.
v. The time between $(j-1)^{th}$ and j^{th} failures depend on the time to the $(j-1)^{th}$ failure.
vi. The initial error content of the software is a random variable.

If $N(t)$, $t \geq 0$, is a counting process indicating the cumulative number of the detected errors within a specified time interval up to a maximum testing duration t, the mean value function denoted by $m(t)$ and the intensity function denoted by $\lambda(t)$ are given below (Yamada *et al.*, 1984):

i. $m(t) = a[1-(1+bt)e^{-bt}$, b > 0, where $m(t)$ is assumed to be non-decreasing with $m(0) = 0$ and $m(\infty) = r$, and r is the total number of errors initially existing within the software.
ii. $\lambda(t) = \alpha\beta^2 te^{-\beta t}$ (2.17)

An adequate SRGM should comply with either S-shaped or exponential SRGM conditions while at the same time yielding reliable predictions (Lee *et al.*, 2014). The S-shaped model derives its name from the shape of the curve resulting from the plot of the cumulative number of the detected errors, and predictions based on this model have been found to be accurate. The NHPP Delayed S-shaped captures the error detection and removal process and assumes a delay between the first failure observation time and the reporting time (Yin & Trivedi, 1999). Most software models assume a perfect case where errors are immediately reported and perfectly removed when detected without introducing new faults (Kaur & Panwar, 2015). Although other software reliability models have been developed to capture the imperfect debugging case, where new faults are introduced into the software and the faults responsible for a failure may not be perfectly removed, the Delayed S-shaped model is unique in that it assumes a delay between the time an error is first observed and when it is reported.

Several researchers have used the Delayed S-shaped software reliability model in software reliability testing. Song *et al.* (2017) used the S-shaped model to develop a new model with an S-shaped growth curve, which can be used in random operating environments to obtain the number of failures detected by time t. The researchers compared the outcome from their new S-shaped model with the other nine NHPP models, such as the GO, Inflection S-shaped, and Yamada imperfect debugging models. They found that their new model fitted the data more significantly than any other NHPP model.

Yin and Trivedi (1999) noted that the Delayed S-shaped software reliability model captures software error removal processes, consisting of two phases: error detection and error isolation. Their research focused on estimating unknown parameters of the GO and Delayed S-shaped models using both the MLE and Bayesian approaches. They used a non-informative prior given by $\frac{1}{\alpha}$, which reduces to zero as α goes to infinity ($\alpha \rightarrow \infty$). The prior was chosen to be consistent with the choice for the GO model. The argument for the non-informative prior was that due to the behaviour of the GO model's log-likelihood function, a prior that reduces to zero as α tends to infinity was appropriate.

In the study by Lee *et al.* (2014), the two unknown parameters of the Delayed S-shaped model, α and β, were assumed to be independently gamma-distributed random variables, yielding a joint informative prior, which was also gamma-distributed. The researchers used the Bayesian approach to predict software reliability and expected testing costs, determining an optimal release time for software systems. The cost estimation method accounted for five costs in software release: set-up cost, routine cost, cost of removing all errors during testing, loss due to a software failure, and opportunity cost. They also performed sensitivity and risk analyses to investigate the impact of misjudgement on prior moments of $E(a)$, $\sigma(a)$, and $E(b)$ on the expected testing cost and reliability. The study revealed that increasing $E(b)$ would improve reliability and reduce expected cost. However, it would attract more costs related to training testing staff or hiring more experienced workers, and thus, it should be thoroughly considered before implementation.

Hanagal and Bhalerao (2018) analysed the Delayed S-shaped SRGM with a time-dependent fault content rate. They proposed three models with different fault content rate functions based on the Delayed S-shaped model: linear, quadratic, and exponential fault content rate functions. The proposed models incorporated error generation over time, capturing a phenomenon in which new errors are introduced into the system while correcting the original faults. The MLE method was used to estimate the unknown parameters of the models using three different data sets. The study compared the new models with the existing Delayed S-shaped model based on Akaike's information criteria, mean sum of squares, error sum of squares, and predictive ratio risk. The study revealed that the new models performed better than the existing Delayed S-shaped model.

In other studies, Gupta *et al.* (2011) used Yamada Delayed S-shaped SRGM in software reliability estimation, incorporating debugging time lag, fault dependency, and imperfect debugging. They found that software reliability improves under imperfect debugging. Anniprincy and Sridhar (2014) proposed a two-dimensional S-shaped model with imperfect debugging. They used logistic function to incorporate the impact of imperfect debugging and error. The study assessed model accuracy using accuracy of estimate (AE) and mean squared error (MSE), comparing the new model with Yamada Rayleigh and Huang Logistic models. The AE was very high, and MSE was lower for the new model than the previous models considered for comparison, indicating its effectiveness.

2.7 Gibbs Sampling

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) algorithm used when direct sampling is difficult. Although the Bayesian method is efficient in estimation, the posterior distribution may often be intractable, making it impossible to perform exact sampling (Dodwell *et al.*, 2015). Moreover, the high dimensionality of the posterior distribution may require focusing on the marginal posterior distribution of each parameter obtained by integrating out over the other parameters in the model (van de Schoot *et al.*, 2021). One way to circumvent this problem is to use the Gibbs sampling technique, which requires obtaining full conditional distributions for each parameter (Martino *et al.*, 2018). The technique works in such a manner that it generates posterior samples for each variable (parameter) by sampling from its full conditional distribution (Bai, 2005; Martino *et al.*, 2018). The technique is used provided the posterior distribution is of a known form. Kuo *et al.* (1997) used the Metropolis algorithm with the Gibbs sampling technique on the GO model to perform predictive analysis. Lee *et al.* (2014) and Bai *et al.* (2005) also used the Gibbs sampling technique to sample from the full conditional distributions and make Bayesian inferences.

2.8 Metropolis-Hastings Technique

The Metropolis-Hastings (MH) is an MCMC algorithm used to sample from distributions that may be difficult to sample from directly (Azzolini *et al.*, 2019). The algorithm is used when the full conditionals obtained do not warrant the use of the Gibbs sampling technique because they do not converge to a known distribution. If at least one of the full conditional distributions is of unknown form (difficult to sample from directly), samples can be generated using a MH-type MCMC method. The MH MCMC approach consists of two steps (Dodwell *et al.*, 2015): (i) generate a new sample given the previous sample using some proposal distribution; (ii) compare the likelihood of the new sample to that of the previous sample to accept (use it for inference) or reject (use the previous sample again) the proposed sample. The MH algorithm can be used within the Gibbs sampling technique to sample from the posterior distributions if some full conditional distributions of the model parameters are of known form.

CHAPTER THREE
MATERIALS AND METHODS

3.1 Research Design

The study focused on developing procedures to obtain Bayesian credible and Wald confidence intervals for the parameters of the Delayed S-shaped model and addressed four issues in one-sample prediction associated with software development testing. A simulation study was performed to construct Bayesian credible and Wald confidence intervals and launch comparisons based on interval lengths and coverage probabilities. The study addressed the following four issues in single-sample software reliability testing:

i. Suppose that the predetermined target value, λ_{tv}, for the software failure rate is not achieved at time *T*, what is the probability that the target value will be achieved at time $\tau, \tau > T$?

ii. Suppose that the target value, λ_{tv}, for the software failure rate is not achieved at time *T*, how long will it take so that the software failure rate will be attained at λ_{tv}?

iii. What is the probability that at most k software failures will occur in the failure time period $(T, \tau]$, $\tau > T$?

iv. What is the Bayesian upper prediction limit (UPL) of $\lambda(\tau) = \alpha\beta^2\tau e^{-\beta\tau}$ with level λ, τ being a predetermined value greater than T?

The study used Bayesian approaches based on informative and non-informative priors to develop posterior and predictive distributions for estimating model parameters and deriving explicit solutions to the above four prediction issues. A gamma-distributed informative prior was adopted, assuming that α and β are independent and distributed as $Gamma(a, b)$ and $Gamma(c, d)$, respectively. Non-informative priors, including $1/\alpha$ and $1/\alpha\beta$ were also used in the analyses.

3.2 Source of Data

Secondary software failure data was used to illustrate the methodologies developed in this study for single-sample prediction. The data was obtained by Ehrlich *et al.* (1993) and also used by Hung-Cuong and Quyet-Thang (2015), with the latter stating that the data has been widely used in the analysis and assessment of software reliability models. The software failure data was in the form of software failure times and inter-failure times, extracted during project T, developed by the American Telephone and Telegraph (AT&T) company (Ehrlich *et al.*, 1993). Other data sets were simulated using the Delayed S-shaped model's intensity function for Bayesian and Wald interval estimation.

3.3 Data Simulation

The data for the analysis was simulated using the thinning algorithm, a simple and appropriate technique for the simulation of an NHPP assuming any form. The simulation was performed by fixing the scale parameter at α = 20 and the shape parameter β = 0.05. The thinning algorithm for simulating inter-failure times for an NHPP, for this case, the Delayed S-shaped software reliability model, is given by Lewis and Shedler (1979).

Step 1: Set $t = 0$, $I = 0$

Step 2: Generate a random number U

Step 3: $t = t - \frac{1}{\lambda}\log \mathrm{U}$. If $t > T$, stop.

Step 4: Generate a random number U_1

Step 5: If $U_1 \le \frac{\lambda(t)}{\lambda}$, set $\mathrm{I} = \mathrm{I} + 1, S(I) = t$

Step 6: Go to step 2.

where $\lambda(t)$ is the model's intensity function given by equation (17), $\lambda = \lambda_0$, $\lambda(t) \le \lambda$, the final value of I represents the number of events in time T, and $S(1), \dots, S(I)$ are the event times. It was assumed that the simulation process emulates the end-user environment and can generate inter-failure times data for reliability testing. An R code was developed to simulate inter-failure times data when T = $100h$, $\lambda_0 = 0.4$, $\alpha_0 = 20$, and $\beta_0 = 0.05$, where λ_0 is the error content rate at time $\mathrm{t} = 0$. A simulation study was performed to compute Bayesian and Maximum Likelihood estimates and construct interval estimates. A program was written such that 5000 samples of different sizes were simulated and used for Bayesian and Wald interval estimation to allow for comparison.

3.4 Wald Confidence Intervals

The Maximum Likelihood estimates of α and β were obtained by maximizing the log-likelihood function. Suppose that $t_1, t_2, \dots, t_n$ denote the n observed failure times in the interval $(0, t]$. Then, the likelihood function for the Delayed S-shaped software reliability model with intensity function in equation (2.17) is given by:

$$L(\alpha, \beta|\underline{t}) = e^{-m(t)} \prod_{i=1}^{n} \lambda(t_i)$$
$$= e^{-\alpha(1-(1+\beta T)\exp(-\beta T)} \prod_{i=1}^{n} (\alpha\beta^2 t_i e^{-\beta t_i})$$
$$= \alpha^n \beta^{2n} (\prod_{i=1}^{n} t_i) e^{-\beta \sum t_i} e^{-\alpha(1-(1+\beta T)\exp(-\beta T))} \qquad \alpha, \beta > 0 \tag{3.1}$$

The log-likelihood function is given by:

$$l(\alpha, \beta|\underline{t}) = nlog\alpha + 2nlog\beta + \log(\prod_{i=1}^{n} t_i) - \beta \sum t_i - \alpha(1 - (1 + \beta T)\exp(-\beta T)) \tag{3.2}$$

The procedure used to construct Wald confidence intervals is described as follows: The log-likelihood function $l(\alpha, \beta/\underline{t})$ was differentiated partially with respect to α and β, and the resulting derivative was equated to zero. The following two equations were obtained:

$$\frac{dl(\alpha, \beta|\underline{t})}{d\alpha} = \frac{n}{\alpha} - \{1 - (1 + \beta T)e^{-\beta T}\}$$
$$\frac{n}{\alpha} - \{1 - (1 + \beta T)e^{-\beta T}\} = 0 \tag{3.3}$$

and;

$$\frac{dl(\alpha, \beta|\underline{t})}{d\beta} = \frac{2n}{\beta} - (\sum_{i=1}^{n} t_i) - \alpha\beta T^2 e^{-\beta T}$$

$$\frac{2n}{\beta} - (\sum_{i=1}^{n} t_i) - \alpha\beta T^2 e^{-\beta T} = 0 \tag{3.4}$$

Equations (3.3) and (3.4) were then simultaneously solved numerically to obtain the Maximum Likelihood estimates $\hat{\alpha}$ and $\hat{\beta}$ of the parameters α and β. The asymptotic variances of the estimates of α and β were obtained from the inverse of the observed Fisher Information matrix, $I(\hat{\alpha}, \hat{\beta})$, which is the matrix containing negative second-order derivatives of the log-likelihood function in equation (3.2).

$$I(\hat{\alpha}, \hat{\beta}) = -E_{(\alpha,\beta)} \begin{bmatrix} \frac{\partial^2 logL(\alpha, \beta|\underline{t})}{\partial\alpha^2} & \frac{\partial^2 logL(\alpha, \beta|\underline{t})}{\partial\alpha\partial\beta} \\ \frac{\partial^2 logL(\alpha, \beta|\underline{t})}{\partial\alpha\partial\beta} & \frac{\partial^2 logL(\alpha, \beta|\underline{t})}{\partial\beta^2} \end{bmatrix}_{(\alpha,\beta)=(\hat{\alpha},\hat{\beta})} \tag{3.5}$$

where;

$$\frac{\partial^2 logL(\alpha, \beta|\underline{t})}{\partial\alpha^2} = -\frac{n}{\alpha^2} \tag{3.6}$$

$$\frac{\partial^2 logL(\alpha, \beta|\underline{t})}{\partial\beta^2} = -\frac{2n}{\beta^2} - \alpha(1 + \beta T)T^2 e^{-\beta T} \tag{3.7}$$

$$\frac{\partial^2 logf(\alpha, \beta|\underline{t})}{\partial\alpha\partial\beta} = -\beta T^2 e^{-\beta T} \tag{3.8}$$

Suppose that the inverse of the observed Fisher information $I(\hat{\alpha}, \hat{\beta})$ in equation (3.5) is given as:

$$\mathrm{I}(\hat{\alpha},\hat{\beta})^{-1} = \begin{bmatrix} \hat{\sigma}_{11}(\hat{\alpha},\hat{\beta}) & \hat{\sigma}_{12}(\hat{\alpha},\hat{\beta}) \\ \hat{\sigma}_{21}(\hat{\alpha},\hat{\beta}) & \hat{\sigma}_{22}(\hat{\alpha},\hat{\beta}) \end{bmatrix} \tag{3.9}$$

Therefore, the $100(1-\alpha)\%$ Wald confidence interval for α was constructed as follows:

$$\hat{\alpha} \pm Z_{\frac{\alpha}{2}}\left[\hat{\sigma}_{11}(\hat{\alpha},\hat{\beta})\right]^{\frac{1}{2}} \tag{3.10}$$

Similarly, the $100(1-\alpha)\%$ Wald confidence interval for β was constructed as follows:

$$\hat{\beta} \pm Z_{\frac{\alpha}{2}}\left[\hat{\sigma}_{22}(\hat{\alpha},\hat{\beta})\right]^{\frac{1}{2}} \tag{3.11}$$

3.5 Bayesian Credible Intervals

A credible interval is a range of values within which an unobserved parameter of interest lies with a specified probability. The Bayesian credible interval of a parameter is computed by first defining its prior function, obtaining posterior distribution, and sampling the parameter from the posterior. The Bayes rule outlines that the accurate posterior estimate (posterior distribution) is obtained by combining prior knowledge with the likelihood function (Petzschner *et al.,* 2015). The prior knowledge about the model parameters forms the prior distribution and is combined with the joint probability of the observed data given model parameters, as presented in equation (3.12). For a model with a single parameter, the data is sampled directly from the posterior distribution or using MCMC techniques. The credible interval is then constructed using quantiles. For a multiparameter model, each parameter is sampled conditioned on the values of others. If the posterior distribution is difficult to sample from directly, MCMC techniques such as Gibbs sampling and Metropolis-Hastings are used. Once the parameter data is generated, histograms are constructed, and credible intervals are obtained as quantiles. For instance, a 95% Bayesian credible interval would be obtained as the 2.5 and 97.5 percentiles of the posterior distribution, while a 90% credible interval would be the 5^{th} and 95^{th} percentiles. The credible intervals are only optimal when the posterior is approximately symmetric, implying the value of γ is split equally to both tails of the marginal posterior distribution of each parameter.

$$\pi(\underline{\theta}/\underline{x}) \propto \pi(\underline{\theta}) * L(\underline{\theta}/\underline{x}) \tag{3.12}$$

where $\pi(\underline{\theta}/\underline{x})$ is the posterior distribution, $\pi(\underline{\theta})$ is the prior function (additional knowledge, experience), and $L(\underline{\theta}/\underline{x})$ is the likelihood function.

3.5.1 Gamma-Distributed Informative Prior

The study adopted the joint gamma-distributed prior for α and β, with the parameters independently distributed as *Gamma*(a,b) and *Gamma*(c,d), respectively. The joint prior is given by:

$$\pi(\alpha,\beta) \propto \alpha^{a-1}\beta^{c-1}e^{-(b\alpha+d\beta)}, \alpha > 0, \beta > 0 \tag{3.13}$$

where *a, b, c*, and *d* are hyper-parameters.

Given the vector of observed software failure times $\underline{t}$, the corresponding joint posterior distribution for α and β was obtained using equation (3.12) as follows:

$$\pi(\alpha,\beta|\underline{t}) \propto \pi(\alpha,\beta)L(\alpha,\beta|\underline{t})$$

$$\propto \alpha^{a-1}\beta^{c-1}e^{-(b\alpha+d\beta)}\,\alpha^{n}\beta^{2n}\left(\prod_{i=1}^{n} t_i\right)e^{-\beta\sum t_i}e^{-\alpha(1-(1+\beta T)\exp(-\beta T)}$$

$$\propto \alpha^{n+a-1}\beta^{2n+c-1}e^{-\alpha[b+1-(1+\beta T)e^{-\beta T}]-\beta[d+\sum_{i=1}^{n} t_i]} \tag{3.14}$$

Full conditional distributions were derived from the joint posterior distribution using the following formulas:

$$h(\alpha|\beta,\underline{t}) = \frac{\pi(\alpha,\beta|\underline{t})}{\int \pi(\alpha,\beta|\underline{t})d\alpha} \tag{3.15}$$

$$h(\beta|\alpha,\underline{t}) = \frac{\pi(\alpha,\beta|\underline{t})}{\int \pi(\alpha,\beta|\underline{t})d\beta} \tag{3.16}$$

The full conditional for alpha was obtained using equation (3.15) as:

$$h(\alpha|\beta,\underline{t}) \propto \alpha^{n+a-1}e^{-\alpha\left(b+1-(1+\beta T)e^{-\beta T}\right)} \tag{3.17}$$

where equation (3.17) is a kernel of gamma distribution with shape parameter $(n+a)$ and scale parameter $\left(b+1-(1+\beta T)e^{-\beta T}\right)$.

Similarly, using equation (3.16), the full conditional for β was obtained as:

$$h(\beta|\alpha,\underline{t}) \propto \beta^{2n+c-1}e^{\alpha(1+\beta T)e^{-\beta T}}e^{-\beta[d+\sum_{i=1}^{n} t_i]} \tag{3.18}$$

which is of unknown distribution.

3.5.2 Prior Function $1/\alpha\beta$

Equations (3.1) and (3.12) were used with the prior function $1/\alpha\beta$ to obtain posterior distribution as follows:

$$\pi(\alpha,\beta|\underline{t}) \propto \frac{1}{\alpha\beta}\,\alpha^{n}\beta^{2n}\left(\prod_{i=1}^{n} t_i\right)e^{-\beta\sum t_i}e^{-\alpha(1-(1+\beta T)\exp(-\beta T))}$$

$$\propto \alpha^{n-1}\beta^{2n-1}e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]-\beta\sum_{i=1}^{n}t_i} \quad (3.19)$$

The full conditional distributions were obtained using equations (3.15) and (3.16) as follows:

$$h(\alpha|\beta,\underline{t}) \propto \alpha^{n-1}e^{-\alpha\left(1-(1+\beta T)e^{-\beta T}\right)} \quad (3.20)$$

which is a kernel of gamma distribution with shape parameter, *n*, and scale parameter $(1-(1+\beta T)e^{-\beta T})$.

$$h(\beta|\alpha,\underline{t}) \propto \beta^{2n-1}e^{\alpha(1+\beta T)e^{-\beta T}}e^{-\beta\sum_{i=1}^{n}t_i} \quad (3.21)$$

which is of unknown distribution.

3.5.3 Prior Function $1/\alpha$

For this prior, the posterior density function was obtained using equations (3.1) and (3.12) as follows:

$$\pi(\alpha,\beta|\underline{t}) \propto \frac{1}{\alpha}.\alpha^{n}\beta^{2n}\left(\prod_{i=1}^{n}t_i\right)e^{-\beta\sum t_i}e^{-\alpha(1-(1+\beta T)\exp(-\beta T))}$$

$$\propto \alpha^{n-1}\beta^{2n}e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]-\beta\sum_{i=1}^{n}t_i} \quad (3.22)$$

The following full conditional distributions were obtained from the posterior density function using equations (3.15) and (3.16).

$$h(\alpha|\beta,\underline{t}) \propto \alpha^{n-1}e^{-\alpha\left(1-(1+\beta T)e^{-\beta T}\right)} \quad (3.23)$$

which is a kernel of gamma distribution with shape parameter, *n*, and scale parameter $\left(1-(1+\beta T)e^{-\beta T}\right)$.

$$h(\beta|\alpha,\underline{t}) \propto \beta^{2n}e^{\alpha(1+\beta T)e^{-\beta T}}e^{-\beta\sum_{i=1}^{n}t_i} \quad (3.24)$$

which is of unknown distribution.

3.6 Gibbs Sampler

The Gibbs sampling technique is used when it is difficult to sample directly from the posterior distribution and the full conditional distributions are of known form. The following is the Gibbs sampling procedure (Johannes & Polson, 2010):

Step 1: Set initial values $(\alpha_0,\ \beta_0)$

Step 2: Sample $\alpha_1 \sim \pi\left(\alpha\middle|\beta_0,\underline{t}\right)$ to have the current state as $(\alpha_1,\ \beta_0)$

Sample $\beta_1 \sim \pi\left(\beta\middle|\alpha_1,\underline{t}\right)$ to have the current state as $(\alpha_1,\ \beta_1)$

Step 3: Sample $\alpha_2 \sim \pi(\alpha|\beta_1, \underline{t})$ to have the current state as $(\alpha_2,\ \beta_1)$

Sample $\beta_2 \sim \pi(\beta|\alpha_2, \underline{t})$ to have the current state as $(\alpha_2,\ \beta_2)$

Step 4: Repeat steps 2 and 3 *n* times, that is, until the desired sample size is obtained.

3.7 Metropolis-Hastings Sampling Technique

The technique is used when it is difficult to sample directly from the posterior, and the full conditional distributions are of unknown form. The MH algorithm is described as follows (Johannes & Polson, 2010; Robert *et al.*, 2004):

Let θ be the parameter of interest, α or β.

Step 1: Let the proposal distribution be $q(.|\underline{t})$

Step 2: Set the initial state θ_0

Step 3: For $n = 1, \ldots, N$;

Generate $\theta \sim q(./\theta_{n-1})$

Step 4: Compute the probability (ratio):

$$A(\theta|\theta_{n-1}) = \min\left\{\frac{q(\theta_{n-1}/\theta)\pi(\theta/\underline{t})}{q(\theta/\theta_{n-1})\pi(\theta_{n-1}/\underline{t})}, 1\right\} \tag{3.25}$$

Step 5: Draw $u \sim u(0,1)$

If $u \leq A(\theta|\theta_{n-1})$, $\theta_n = \theta$

Else, $\theta_n = \theta_{n-1}$

If the two distributions from the proposal distribution are symmetric, the ratio becomes:

$$A(\theta|\theta_{n-1}) = \min\left\{\frac{\pi(\theta/\underline{t})}{\pi(\theta_{n-1}/\underline{t})}, 1\right\} \tag{3.26}$$

3.8 Metropolis-Hastings within Gibbs

It was difficult to sample directly from the derived joint posterior distributions given by equations (3.14), (3.19), and (3.22). A possible way of simulating these posterior densities was by using their corresponding full conditional distributions for each parameter α and β via Gibbs sampling technique. However, the full conditional distributions $\pi(\beta/\alpha, \underline{t})$ for β

were improper for the three Bayesian approaches, implying it was difficult to sample from them directly. As such, the MH within Gibbs MCMC approach was used. The following algorithm was used.

Step 1: Start with $k = 1$ and set the initial values of $\{\underline{\theta}^{(1)} = (\alpha^{(1)}, \beta^{(1)})\}$

Step 2: Sample α_1 from $\alpha_k \sim p(\alpha|\beta_{k-1}, \underline{t})$ (which is gamma-distributed) to have the current state as $(\alpha_1,\ \beta_0)$.

Step 3: Using the proposal distribution of β, where the proposal was chosen as $\beta \sim N(\beta^{(k-1)}, \sigma_\beta^2)$, sample a candidate value for β^{prop} using α_1 obtained in step two.

Step 4: Generate $u \sim UNIF(0,1)$

Step 5: Compute the Metropolis-Hastings acceptance ratio at the candidate value, $\underline{\theta}^{prop}$, and the previous value, $\underline{\theta}^{(k-1)}$, using block updating.

$$R_{\underline{\theta}} = \frac{\pi_{\underline{\theta}}(\underline{\theta}^{prop} | \underline{t}) * q_{\underline{\theta}}(\underline{\theta}^{(k-1)} | \underline{\theta}^{prop})}{\pi_{\underline{\theta}}(\underline{\theta}^{(k-1)} | \underline{t}) * q_{\underline{\theta}}(\underline{\theta}^{prop} | \underline{\theta}^{(k-1)})} \tag{3.27}$$

Step 6: If $u \le \min(1,\ R_{\underline{\theta}})$, accept the candidate point, β_k, with probability $min\ (1,\ R_{\underline{\theta}})$: set $\underline{\theta}^{(k)} = \underline{\theta}^{prop}$. Otherwise set $\underline{\theta}^{(k)} = \underline{\theta}^{(k-1)}$ to have the current state $(\alpha_1,\ \beta_1)$.

Step 7: Repeat steps 2 to 6 until the desired sample size, *k*, is obtained.

Based on the simulation of the joint posterior distribution of α and β, histograms were constructed, and $100(1-\alpha)\%$ credible intervals for the two parameters were computed as follows: Let $\gamma \in (0,1)$. If β_L^* and β_U^* are respectively the $\frac{\gamma}{2}$ and $\left(1-\frac{\gamma}{2}\right)$ posterior quantiles for β, then $(\beta_L^*,\ \beta_U^*)$ is a $100(1-\gamma)\%$ credible interval for β. Similarly, if α_L^* and α_U^* are respectively the $\frac{\gamma}{2}$ and $\left(1-\frac{\gamma}{2}\right)$ posterior quantiles for α, then $(\alpha_L^*,\ \alpha_U^*)$ is a $100(1-\gamma)\%$ credible interval for α.

3.9 Comparison

Interval lengths and coverage probabilities were used for comparison. Samples of inter-failure times data of different sizes were simulated from the intensity function of the NHPP Delayed S-shaped model. Each sample was used to compute the Wald confidence and Bayesian credible intervals for the model parameters, *α* and *β*. For each interval estimate, coverage probability was obtained as the ratio of the number of intervals that contained the true parameter value to the total number of samples considered. Interval lengths were computed for each interval estimate as the difference between the upper and lower bounds. The procedure that yielded a shorter interval length and a higher coverage probability was considered more precise.

3.10 Predictive Analysis

Bayesian predictive analysis on the model was performed with each of the three prior functions. The posterior distributions obtained were used to derive the posterior predictive distribution using the formula in equation (3.28). The methodologies for addressing the four prediction issues were also derived.

$$f\left(t_{n+k}/\underline{t}\right) = \int f(t_{n+k}/\theta)\pi\left(\theta/\underline{t}\right)d\theta \qquad (3.28)$$

where $\pi\left(\theta/\underline{t}\right)$ is the posterior density function, $f\left(t_{n+k}/\underline{t}\right)$ is the predictive density function of $(n+k)^{th}$ future failure time and θ is the parameter of interest, in this case, α and β.

3.11 Data Analysis

The methods derived were illustrated using secondary data and simulated software failure data. The posterior distributions and predictive inferences for the simulation study and single sample cases, using informative and non-informative priors, lacked closed-form solutions. As such, the MH within Gibbs sampling technique was used to sample from the posterior. The analyses were performed using R-Language version 4.3.0.

CHAPTER FOUR

RESULTS AND DISCUSSION

This section discusses the results of this study in two parts. The first part presents the results of the simulation study on the construction of Bayesian credible intervals and compares them with Wald confidence intervals. The second part presents the derivations of methodologies addressing the four prediction issues, which are outlined as propositions in this section. Proofs of these propositions are also provided, and an illustration using secondary software failure data was also performed.

4.1 Bayesian and Wald Interval Estimation

The study investigated the performance of the Wald and Bayesian interval estimation methods based on data simulated from the Delayed S-shaped NHPP model's intensity function. It was assumed that the simulation process emulates the end-user environment and generates inter-failure times data for reliability testing. An R code was developed to simulate inter-failure times data when T = 100*h*, $\lambda_0 = 0.4$, $\alpha_0 = 20$, and $\beta_0 = 0.05$, where λ_0 is the error content rate at time t = 0.

4.1.1 Wald Confidence Intervals and Bayesian Credible Intervals

The Wald confidence intervals and Bayesian credible intervals were constructed using the developed methodologies, where results generated using two samples of sizes 15 and 33 were reported in Table 4.1 for illustrations. The sample sizes were chosen such that 15 represents a small sample while 33 represent a large sample. These were chosen arbitrarily to show that the Bayesian credible intervals were successfully constructed using informative and non-informative priors. The credible intervals indicate a range of values that contain the true parameter value with a 95% probability. For instance, the 95% credible interval for α obtained using the Bayesian method with gamma-distributed informative prior indicates that there is a 95% probability that the true value of α will lie within 11.5417 and 26.6372.

Table 4.1: Wald confidence and Bayesian credible intervals obtained using different sample sizes.

Sample size	Method	Parameter	Confidence/ Credible Interval	Interval length
15	Wald	α	$(7.6789, 25.8552)$	18.1763
		β	$(0.0174, 0.0591)$	0.0417
	Bayesian (Informative)	α	$(11.5417, 26.6372)$	15.0956
		β	$(0.0222, 0.0587)$	0.0364
	Bayesian $(1/\alpha\beta)$	α	$(8.9138, 25.3769)$	16.4651
		β	$(0.0325, 0.0663)$	0.0337
	Bayesian $(1/\alpha)$	α	$(8.8226, 25.2418)$	16.4142
		β	$(0.0328, 0.0660)$	0.0332
33	Wald	α	$(22.7212, 46.9662)$	24.2450
		β	$(0.0319, 0.0616)$	0.0296
	Bayesian (Informative)	α	$(21.5746, 39.5105)$	17.9359
		β	$(0.0347, 0.0616)$	0.0270
	Bayesian $(1/\alpha\beta)$	α	$(23.4467, 46.9717)$	23.5251
		β	$(0.0412, 0.0663)$	0.0251
	Bayesian $(1/\alpha)$	α	$(23.5633, 46.9573)$	23.394
		β	$(0.0409, 0.0669)$	0.0260

4.1.2 Comparison

The simulated inter-failure times data were used to sample α and β random variables from the posterior distributions using the Metropolis Hasting within Gibbs MCMC procedure. A normal distribution was proposed for β, and the unknown parameter theta was fixed at $\theta = 0.05$. Once the data for α and β were obtained, they were assessed to determine general patterns. The trace plots of the parameters were constructed to examine stability. As shown in Figure 4.1, the blue trace plots were generated using the gamma-distributed informative prior, the red trace plots were constructed using the data from the Bayesian method with $1/\alpha\beta$ prior, and the green trace plots are for $1/\alpha$ prior. The plots show the stability of the sampled parameter values since they contain the true (fixed) values indicated by the horizontal lines and have approximately uniform upward and downward spikes.

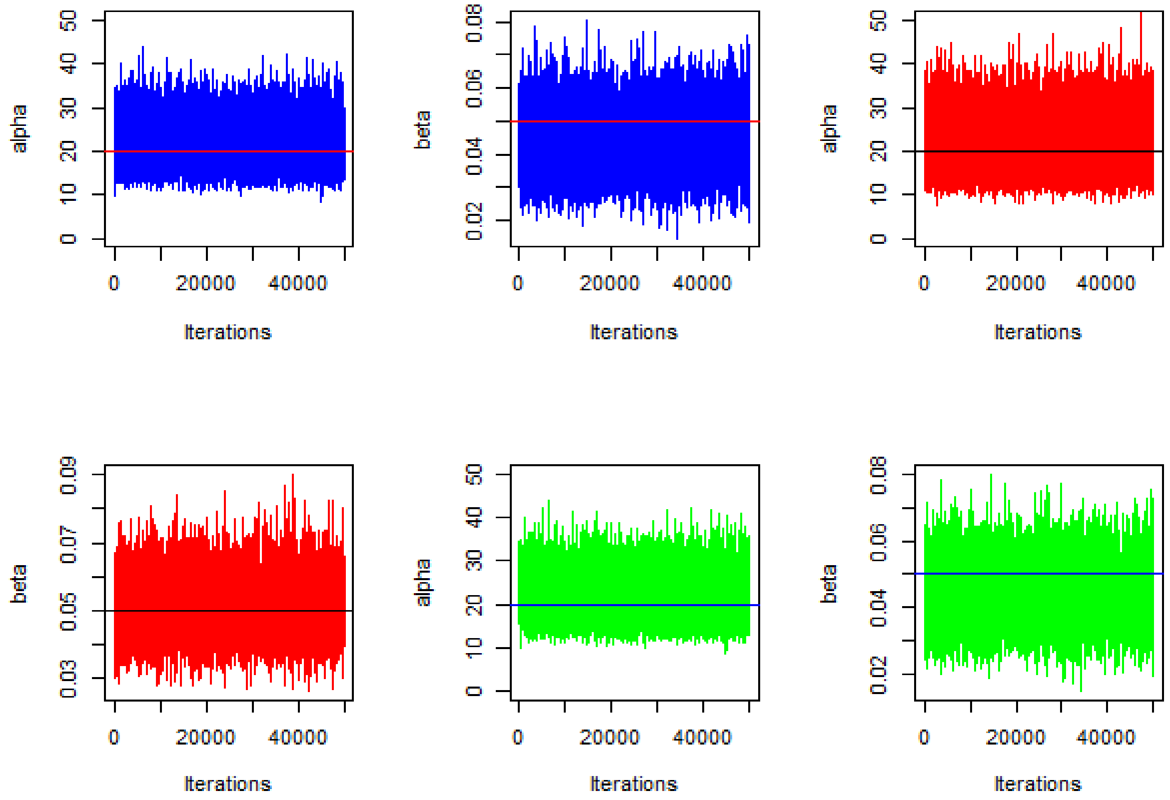

Figure 4.1: Trace plots for α and β generated using gamma distributed informative prior(blue), *1/αβ* non-informative prior (red), and *1/α* prior (green).

The marginal posterior distributions of the two parameters were assessed. Figures 4.2, 4.3, and 4.4 show histograms for α and β constructed using the three Bayesian methods (gamma-distributed informative prior and non-informative priors; $1/\alpha\beta$ and $1/\alpha$). Since the histograms are approximately normally distributed, the value of γ is equally split to both tails of the marginal posterior distribution of each parameter.

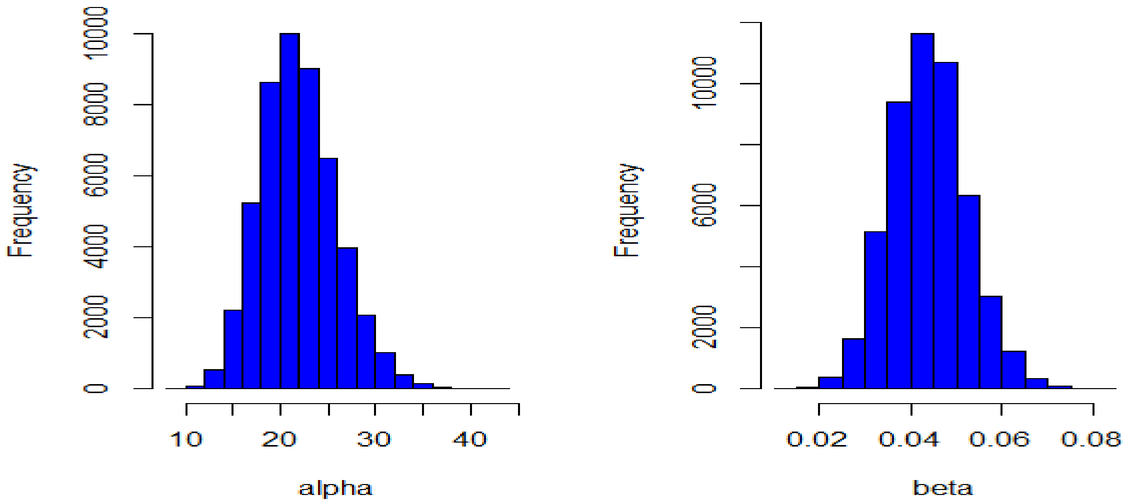

Figure 4.2: Histogram of α and β generated using gamma distributed informative prior.

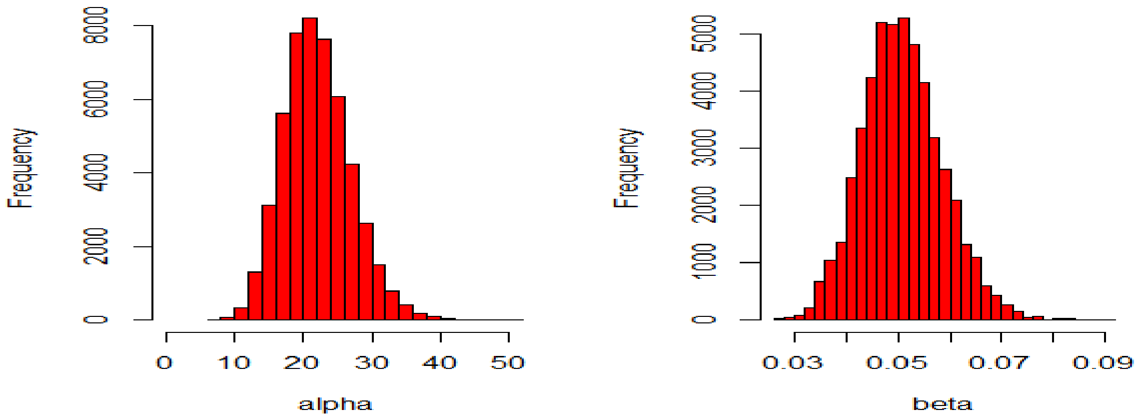

Figure 4.3: Histogram of α and β generated using *1/αβ* non-informative prior.

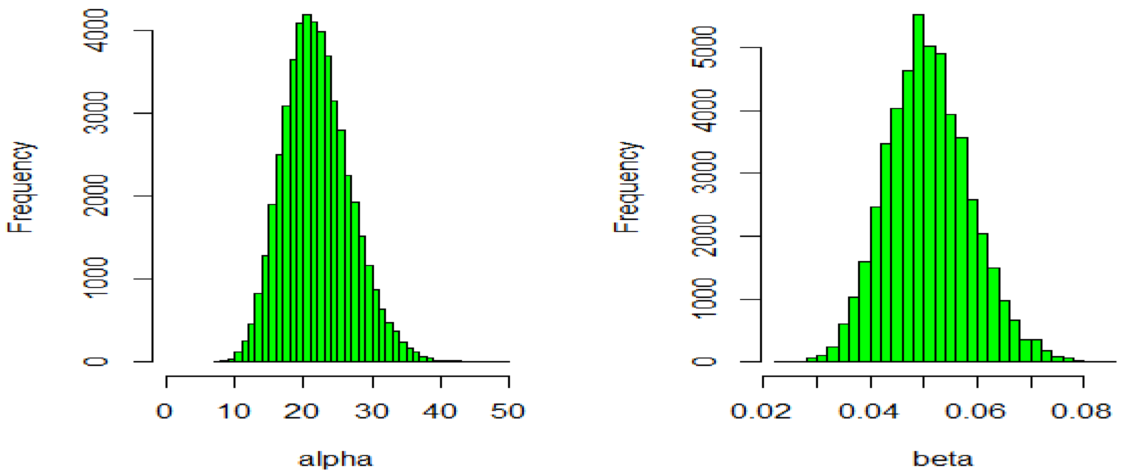

Figure 4.4: Histogram of α and β generated using *1/α* non-informative prior.

A total of 5000 samples of inter-failure times data were simulated and used to generate 5000 samples, each for α and β from the three posterior distributions. The inter-failure times samples had an average size of 23. An R code was developed such that the same 5000 samples of inter-failure times data were used for the four methods (Wald and three Bayesian approaches) for accurate comparisons. The datasets were used to compute Maximum Likelihood estimates of the model's parameters, construct their corresponding confidence intervals, and generate coverage probabilities through repeated sampling. Moreover, Bayesian credible intervals and coverage probabilities were obtained for each sample of α and β random variables generated from the posterior distributions. In this regard, each method had a total of 5000 interval estimates for each parameter. Interval lengths were obtained as the difference between the upper and lower bounds of the interval estimates.

Coverage probabilities for each method were estimated as the proportion of the 5000 confidence and credible intervals that contained the true parameter value. The coverage probabilities were recorded as obtained, while summary statistics (minimum, maximum, mean, and standard deviation) of widths of the 5000 confidence and credible intervals for each method were obtained and presented in Table 4.2. The results indicate that two Bayesian methods (with informative and $1/\alpha\beta$ priors) have shorter average interval lengths than the Wald approach. Of the methods, Bayesian with informative prior was superior, yielding higher coverage probabilities and shorter average widths. The estimated coverage probabilities of Wald confidence and Bayesian credible intervals for α and β were reasonably close to the theoretical value of 0.95.

Table 4.2: Estimated summary statistics and coverage probabilities of widths of the 95% Bayesian credible and Wald confidence intervals for the parameters α and β on the basis of 5000 samples.

Method		Min	Max	Mean	Std dev	C_p
Wald	α	11.40	61.40	18.83	3.7986	0.943
	β	0.02811	0.09064	0.04123	0.00685	0.95
Bayesian (Informative)	α	11.48	19.12	15.06	1.2149	0.988
	β	0.02297	0.07146	0.03665	0.00610	0.960
Bayesian $(\mathbf{1/\alpha\beta})$	α	11.18	25.07	17.79	2.1118	0.95
	β	0.02098	0.08831	0.03627	0.00771	0.934
Bayesian $(\mathbf{1/\alpha})$	α	11.60	855.77	22.50	31.12	0.955
	β	0.02805	0.09369	0.04190	0.00711	0.934

Moreover, trace plots were constructed for the interval lengths of α and β, as shown in Figures 4.5 and 4.6. All the plots were constructed using the same limits on the y-axis for easier comparison. It can be noticed that the interval lengths for the Bayesian credible intervals obtained using the joint gamma-distributed informative prior were shorter than both the Wald and non-informative priors. For the parameter α, the interval lengths generated using the gamma-distributed informative and $1/\alpha\beta$ priors had shorter and uniform downward and upward spikes, as displayed in Figure 4.5, the second and third graphs. The graphs show that the two interval estimation methods have increased precision (see red and green graphs). The first graph in Figure 4.5, generated using the Wald approach, shows long upward spikes but almost uniform downward spikes. However, the fourth graph has long upward spikes, suggesting that $1/\alpha$ prior would yield unreasonably wider credible intervals than Wald and the other two Bayesian methods.

In Figure 4.6, the spikes for the interval lengths for the Wald confidence and Bayesian credible intervals show a similar pattern. The downward spikes for the four trace plots are shorter and tend to be uniform. However, the long upward spikes are fewer in the second graph generated using the Bayesian method with gamma-distributed informative prior. Smaller and slightly different standard deviations for β reported in Table 4.2, column 6, indicate the reason for similar behaviour in the graphs.

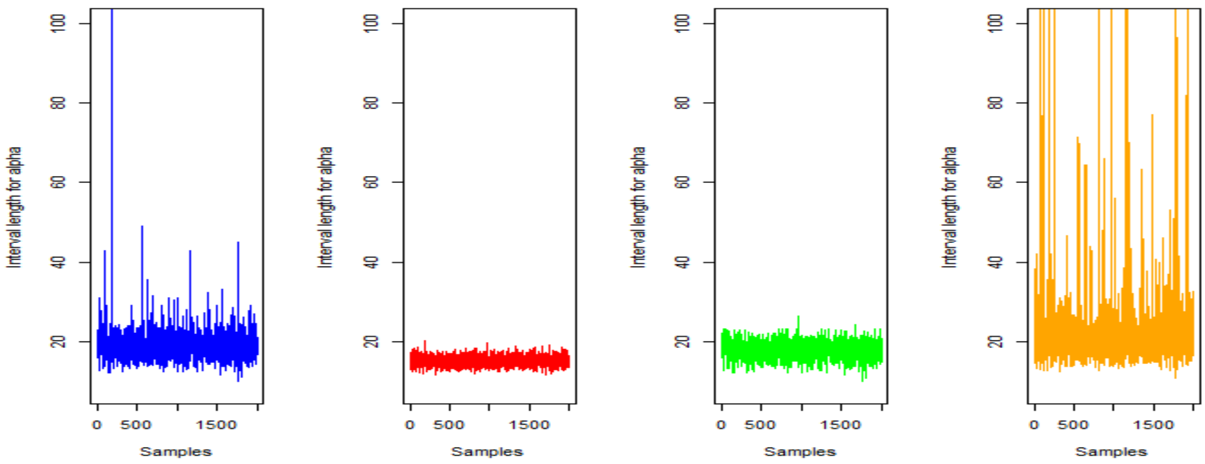

Figure 4.5: Trace plots for interval lengths of α generated through the Wald approach (blue), Bayesian with informative prior (red), *1/αβ* prior (green), and *1/α* prior (orange).

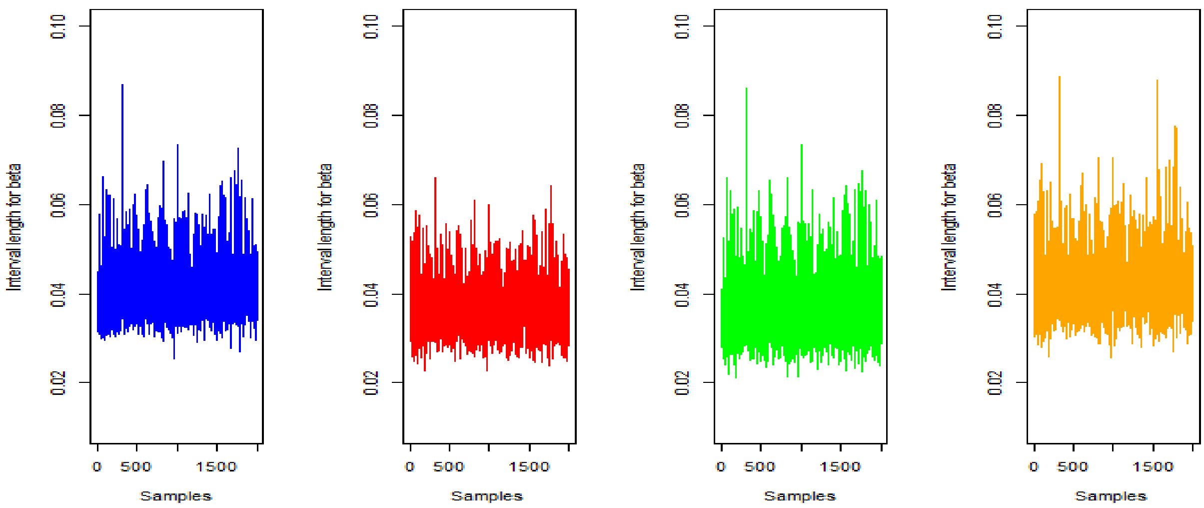

Figure 4.6: Trace plots for interval lengths of β generated through the Wald approach (blue), Bayesian with informative prior (red), *1/αβ* prior (green), and *1/α* prior (orange).

4.2 Predictive Analysis Results

4.2.1 Some Results Used in the Derivation of Methodologies

When testing stops after a predetermined number of failures, *n*, the failure data is said to be failure-truncated, and the n failure times are denoted by $Y_{obs}^{f} = [t_i]_{i=1}^{n}$. However, if testing stops at a predetermined time, *t*, the failure data is said to be time-truncated, and the corresponding observed failure data is denoted by $Y_{obs}^{t} = [n, t_1, \dots, t_n; t]$. Let Y_{obs} represent Y_{obs}^{t} or Y_{obs}^{f}. The joint density of Y_{obs} is obtained as:

$$f(Y_{obs} \mid \alpha,\beta) = \alpha^n\beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]} \quad (4.1)$$

According to Bayes' rule, the posterior density is obtained using the formula:

$$\pi(\theta|\underline{t}) = \frac{\pi(\theta,\underline{t})}{f(\underline{t})}$$

$$= \frac{f(\underline{t}|\theta)\,\pi(\theta)}{f(\underline{t})} \quad (4.2)$$

The posterior predictive distribution of t^+ is given as;

$$f(t^+|\,\underline{t}) = \int f(t^+,\theta|\,\underline{t})\,d\theta$$

$$= \int f(t^+|\theta,\underline{t})\,\pi(\theta|\underline{t})d\theta$$

$$= \int f(t^+|\theta)\pi(\theta|\underline{t})d\theta \quad (4.3)$$

4.2.2 Case 1: Non-Informative Prior Given by $1/\alpha$

When the shape parameter, β, is known, the following non-informative prior distribution for α was used:

$$\pi(\alpha) \propto \frac{1}{\alpha},\ \alpha > 0$$

The posterior distribution of α can be obtained from equation (4.2) as:

$$\pi(\alpha|Y_{obs}) = \frac{f(Y_{obs}|\alpha,\beta)\pi(\alpha,\beta)}{\int_0^\infty\int_0^\infty f(Y_{obs}|\alpha,\beta)\pi(\alpha,\beta)d\alpha d\beta}$$

By substituting equation (4.1) and equation (4.2), the following equation was obtained.

$$\pi(\alpha|Y_{obs}) = \frac{\alpha^{n-1}\beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}}{\int_0^\infty \alpha^{n-1}\beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}d\alpha} \quad (4.4)$$

The denominator simplifies to:

$$\int_0^\infty \alpha^{n-1}\beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}d\alpha$$

$$= \beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}\int_0^\infty \alpha^{n-1}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}d\alpha$$

$$= \beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}\frac{\Gamma(n)}{[1-(1+\beta T)e^{-\beta T}]^n}\int_0^\infty \frac{[1-(1+\beta T)e^{-\beta T}]^n}{\Gamma(n)}\alpha^{n-1}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}d\alpha \quad (4.5)$$

The integrand in equation (4.5) is a gamma distribution with parameters n and $1 - (1+\beta T)e^{-\beta T}$, hence, it integrates to one (1). Thus, the denominator simplifies to:

$$= \frac{\Gamma(n)(\prod_{i=1}^{n} t_i)\beta^{2n}e^{-\beta\sum_{i=1}^{n} t_i}}{\left[1-(1+\beta T)e^{-\beta T}\right]^n}$$

Equation (4.4) becomes:

$$\pi(\alpha|Y_{obs}) = \frac{\alpha^{n-1}\beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]}}{\Gamma(n)(\prod_{i=1}^{n} t_i)\beta^{2n}e^{-\beta\sum_{i=1}^{n} t_i}/\left[1-(1+\beta T)e^{-\beta T}\right]^n}$$

which reduces to:

$$\pi(\alpha|Y_{obs}) = [\Gamma(n)]^{-1}\alpha^{n-1}e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]}\left[1-(1+\beta T)e^{-\beta T}\right]^n \qquad (4.6)$$

Let t^+ be the random variable being predicted. The posterior predictive distribution of t^+ was obtained using equation (4.3) as:

$$f(t^+|Y_{obs}) = \int_0^\infty f(t^+|Y_{obs},\alpha)\pi(\alpha|Y_{obs})\,d\alpha \qquad (4.7)$$

The Bayesian UPL for t^+ with level γ satisfies:

$$\gamma = \int_{-\infty}^{y_U^{(\beta)}} p(t^+|Y_{obs})dt^+ \qquad (4.8)$$

4.2.3 Case 2: Non-Informative Prior Given by $1/\alpha\beta$

When the shape parameter β is unknown, the following non-informative density of α and β was considered, assuming they are mutually independent.

$$\pi(\alpha,\beta) \propto \frac{1}{\alpha\beta}, \qquad \alpha,\beta > 0 \qquad (4.9)$$

The corresponding posterior joint density was obtained using equation (4.2) as follows:

$$\pi(\alpha,\beta|Y_{obs}) = \frac{f(Y_{obs}|\alpha,\beta)\pi(\alpha,\beta)}{\int_0^\infty\int_0^\infty f(Y_{obs}|\alpha,\beta)\pi(\alpha,\beta)d\alpha d\beta}$$

Equations (4.1) and (4.3) were used to obtain:

$$\pi(\alpha,\beta|Y_{obs}) = \frac{\alpha^{n-1}\beta^{2n-1}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]}}{\int_0^\infty\int_0^\infty \alpha^{n-1}\beta^{2n-1}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]}d\alpha d\beta} \qquad (4.10)$$

Simplifying the denominator:

$\int_0^\infty \int_0^\infty \alpha^{n-1}\beta^{2n-1}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}d\alpha d\beta$

$= (\prod_{i=1}^n t_i)\int_0^\infty \beta^{2n-1}\, e^{-\beta\sum_{i=1}^n t_i}\left[\int_0^\infty \alpha^{n-1}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}d\alpha\right]d\beta$

$=$

$(\prod_{i=1}^n t_i)\int_0^\infty \beta^{2n-1}\, e^{-\beta\sum_{i=1}^n t_i}\left(\frac{\Gamma(n)}{\left(1-(1+\beta T)e^{-\beta T}\right)^n}\right)\left[\int_0^\infty \frac{\left(1-(1+\beta T)e^{-\beta T}\right)^n}{\Gamma(n)}\alpha^{n-1}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}d\alpha\right]d\beta$

The last part of the denominator, $\int_0^\infty \frac{\left(1-(1+\beta T)e^{-\beta T}\right)^n}{\Gamma(n)}\alpha^{n-1}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}d\alpha = 1$

Therefore, the denominator reduces to:

$= (\prod_{i=1}^n t_i)\int_0^\infty \frac{\Gamma(n)}{\left(1-(1+\beta T)e^{-\beta T}\right)^n}\beta^{2n-1}\, e^{-\beta\sum_{i=1}^n t_i}d\beta$

$= (\prod_{i=1}^n t_i)\Gamma(n)\int_0^\infty \frac{\beta^{2n-1}e^{-\beta\sum_{i=1}^n t_i}}{\left(1-(1+\beta T)e^{-\beta T}\right)^n}d\beta$

Equation (4.4) reduces to:

$$\pi(\alpha,\beta|Y_{obs}) = \frac{\alpha^{n-1}\beta^{2n-1}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}}{(\prod_{i=1}^n t_i)\Gamma(n)\int_0^\infty \frac{\beta^{2n-1}e^{-\beta\sum_{i=1}^n t_i}}{\left(1-(1+\beta T)e^{-\beta T}\right)^n}d\beta} \quad (4.11)$$

The integral part $\int_0^\infty \frac{\beta^{2n-1}e^{-\beta\sum_{i=1}^n t_i}}{\left(1-(1+\beta T)e^{-\beta T}\right)^n}d\beta$ in the denominator of equation (4.5) has no closed form, and thus, the MCMC numerical technique was used to obtain its value. This integral part was denoted by a constant k as $k = \int_0^\infty \frac{\beta^{2n-1}e^{-\beta\sum_{i=1}^n t_i}}{\left(1-(1+\beta T)e^{-\beta T}\right)^n}d\beta$, reducing equation (4.5) to:

$$\pi(\alpha,\beta|Y_{obs}) = [k\Gamma(n)]^{-1}\alpha^{n-1}\beta^{2n-1}e^{-\beta\sum_{i=1}^n t_i}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]} \quad (4.12)$$

Similar to equation (4.7) and equation (4.8), the posterior predictive distribution becomes:

$$f(t^+|Y_{obs}) = \int_0^\infty \int_0^\infty f(t^+|Y_{obs},\alpha,\beta)\,\pi(\alpha,\beta|Y_{obs})d\alpha d\beta \quad (4.13)$$

and the Bayesian UPL is:

$$\gamma = \int_{-\infty}^{yU} p(t^+|Y_{obs})dt^+ \quad (4.14)$$

4.3 Main Results for Single-Sample Prediction Using Non-Informative Priors

4.3.1 Proposition I

The probability that the target value λ_{tv} will be achieved at time τ $(\tau > T)$ is:

$$\gamma = \begin{cases} 1 - \sum_{h=0}^{n-1} \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}} & if\ \beta\ is\ known \\ 1 - \frac{1}{k}\sum_{h=0}^{n-1}\int_0^\infty \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}} \frac{\beta^{2n-1}e^{-\beta\sum_{i=1}^n t_i}}{\left[1-(1+\beta T)e^{-\beta T}\right]^n} d\beta & if\ \beta\ is\ unknown \end{cases}$$

Proof:

The probability is given by: Let $\pi(Y_{obs}|\lambda_\tau)$ denote the posterior density of $\lambda(t) = \alpha\beta^2 t e^{-\beta t}$. Thus, the probability that λ_{tv} will be achieved at time τ $(\tau > T)$ is given by:

$$\gamma = \Pr\left(\lambda_\tau \le \lambda_{tv} \,\middle|\, Y_{obs}\right)$$

$$= \int_0^{\lambda_{tv}} p(Y_{obs}|\lambda_\tau) d\lambda_\tau \tag{4.15}$$

When $\boldsymbol{\beta}$ is known, $\alpha = \frac{\lambda(t)}{\beta^2 t e^{-\beta t}}$, obtained from the intensity function, and $\frac{d\alpha}{d\lambda_\tau} = \frac{1}{\beta^2 t e^{-\beta t}}$. The posterior distribution of λ_τ is $\pi(\lambda_\tau|Y_{obs}) = \pi(\alpha|Y_{obs})\left|\frac{d\alpha}{d\lambda_\tau}\right|$. Thus, the posterior density becomes:

$$\pi(\lambda_\tau|Y_{obs}) = \frac{1}{\Gamma(n)}\left(\frac{\lambda(t)}{\beta^2\tau e^{-\beta\tau}}\right)^{n-1} e^{-\left(\frac{\lambda(t)}{\beta^2\tau e^{-\beta\tau}}\right)\left[1-(1+\beta T)e^{-\beta T}\right]}\left[1-(1+\beta T)e^{-\beta T}\right]^n . \frac{1}{\beta^2\tau e^{-\beta\tau}}$$

which reduces to:

$$\pi(\lambda_\tau|Y_{obs}) = \frac{\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\right]^n}{\Gamma(n)} \lambda_\tau^{n-1} e^{-\lambda_\tau\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\right]} \tag{4.16}$$

From equation (4.16), λ_τ follows a gamma distribution with parameters n and $\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}$. However, there is a relationship between gamma and Poisson distribution defined by:

$$\frac{\beta^\alpha}{\Gamma(\alpha)}\int_0^\lambda x^{\alpha-1} e^{-\beta x} dx = 1 - \sum_{h=0}^{\alpha-1}\frac{(\beta\lambda)^h}{h!} e^{-\beta\lambda} \tag{4.17}$$

From equation (4.15), $\left(\gamma = \int_0^{\lambda_{tv}} p(Y_{obs} \mid \lambda_\tau) d\lambda_\tau\right)$, and equation (4.17), it follows that:

$$\gamma = 1 - \sum_{h=0}^{n-1} \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}} \tag{4.18}$$

Equation (4.18) shows the first formula in Proposition I.

When β is unknown: From the intensity function, $\alpha = \frac{\lambda(t)}{\beta^2 t e^{-\beta t}}$, and let $\beta = \beta$. The Jacobian is $\frac{d(\alpha,\beta)}{d(\lambda_\tau,\beta)} = \frac{1}{\beta^2 t e^{-\beta t}}$. The joint posterior density of (λ_τ, β) is given by:

$$\pi(\lambda_\tau, \beta \mid Y_{obs}) = \pi(\alpha, \beta \mid Y_{obs}) \left|\frac{d(\alpha,\beta)}{d(\lambda_\tau,\beta)}\right|$$

From equation (4.12):

$$\pi(\lambda_\tau, \beta | Y_{obs}) = [k\Gamma(n)]^{-1} \alpha^{n-1} \beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i} e^{-\alpha[1-(1+\beta T)e^{-\beta T}]} . \frac{1}{\beta^2 \tau e^{-\beta\tau}}$$

$$= [k\Gamma(n)]^{-1} \left(\frac{\lambda_\tau}{\beta^2 \tau e^{-\beta\tau}}\right)^{n-1} \beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i} e^{-\left(\frac{\lambda_\tau}{\beta^2 \tau e^{-\beta\tau}}\right)[1-(1+\beta T)e^{-\beta T}]} . \frac{1}{\beta^2 \tau e^{-\beta\tau}}$$

$$= \left(\frac{1}{\beta^2 \tau e^{-\beta\tau}}\right)^{n-1} \frac{\beta^{2n-1}}{k\Gamma(n)} e^{-\beta \sum_{i=1}^n t_i} \lambda_\tau^{n-1} e^{-\lambda_\tau\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]} \frac{1}{\beta^2 \tau e^{-\beta\tau}}$$

$$= \left(\frac{1}{\beta^2 \tau e^{-\beta\tau}}\right)^{n} \frac{\beta^{2n-1}}{k\Gamma(n)} e^{-\beta \sum_{i=1}^n t_i} \lambda_\tau^{n-1} e^{-\lambda_\tau\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]}$$

$$= \frac{\left(\beta^2 \tau e^{-\beta\tau}\right)^{-n}}{k\Gamma(n)} \beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i} \left\{\frac{\Gamma(n)}{\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]^n}\right\} \frac{\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]^n}{\Gamma(n)} \lambda_\tau^{n-1} e^{-\lambda_\tau\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]}$$

$$= \frac{\beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i}}{k\left[1-(1+\beta T)e^{-\beta T}\right]^n} \frac{\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]^n}{\Gamma(n)} \lambda_\tau^{n-1} e^{-\lambda_\tau\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]} \tag{4.19}$$

Equations (4.15), (4.17), and (4.19) were used to obtain:

$$\gamma = \frac{1}{k} \int_0^\infty \frac{\beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i}}{\left[1-(1+\beta T)e^{-\beta T}\right]^n} \left\{1 - \sum_{h=0}^{n-1} \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}}\right\} d\beta$$

$$= \frac{1}{k} \int_0^\infty \frac{\beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i}}{\left[1-(1+\beta T)e^{-\beta T}\right]^n} \, d\beta -$$

$$\frac{1}{k} \sum_{h=0}^{n-1} \int_0^\infty \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}} \frac{\beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i}}{\left[1-(1+\beta T)e^{-\beta T}\right]^n} \, d\beta$$

But $k = \int_0^\infty \frac{\beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i}}{\left(1-(1+\beta T)e^{-\beta T}\right)^n} d\beta$

$$\gamma = 1 - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^\infty \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}} \lambda_{tv}\right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}} \lambda_{tv}} \frac{\beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i}}{\left[1-(1+\beta T)e^{-\beta T}\right]^n} \, d\beta \tag{4.20}$$

Equation (4.20) is the formula in the second part of Proposition I.

4.3.2 Proposition II

Let τ^* denote the time required to attain λ_{tv}. For a specified level γ:

$$\tau^* = \begin{cases} \left[-\frac{1}{\beta} W_n \left(\frac{-2\lambda_{tv}\left[1-(1+\beta T)e^{-\beta T}\right]}{\beta \chi^2(2n;\gamma)}\right)\right] - T & if\ \beta\ is\ known \\ \tau - T & if\ \beta\ is\ unknown \end{cases}$$

Remark 1:

The second part of proposition II is such that τ is the solution to the equation

$$\gamma = 1 - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^\infty \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}} \lambda_{tv}\right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}} \lambda_{tv}} \frac{\beta^{2n-1} e^{-\beta \sum_{i=1}^n t_i}}{\left[1-(1+\beta T)e^{-\beta T}\right]^n} \, d\beta$$

Proof:

For a given level γ, the time required to attain the target value λ_{tv} is given by $\tau^* = \tau - T$, where τ satisfies equation (4.15). **When $\boldsymbol{\beta}$ is known;** from equation (4.15), it can be noted that $2\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}}\right] \lambda_{tv}$ follows a Chi-square distribution with $2n$ degrees of freedom. Therefore;

$$2\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}}\right] \lambda_{tv} = \chi^2(2n;\gamma) \tag{4.21}$$

$$\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}} = \frac{\chi^2(2n;\gamma)}{2\lambda_{tv}} \tag{4.22}$$

$$\beta^2 \tau e^{-\beta \tau} = \frac{2\lambda_{tv}\left[1-(1+\beta T)e^{-\beta T}\right]}{\chi^2(2n;\gamma)}$$

$$\tau e^{-\beta \tau} = \frac{2\lambda_{tv}\left[1-(1+\beta T)e^{-\beta T}\right]}{\beta^2 \chi^2(2n;\gamma)}$$

$$\tau = -\frac{1}{\beta} W_n \left(\frac{-2\beta\lambda_{tv}\left[1-(1+\beta T)e^{-\beta T}\right]}{\beta^2 \chi^2(2n;\gamma)}\right)$$

$$= -\frac{1}{\beta} W_n \left(\frac{-2\lambda_{tv}\left[1-(1+\beta T)e^{-\beta T}\right]}{\beta \chi^2(2n;\gamma)} \right), \text{ for } n \in \text{Z} \tag{4.23}$$

where $n \in$ Z denotes the n^{th} root of the equation $\tau e^{\tau} = \frac{-2\lambda_{tv}\left[1-(1+\beta T)e^{-\beta T}\right]}{\beta \chi^2(2n;\gamma)}$, and W is the Lambert W function, satisfying $W(\tau e^{\tau}) = \tau$.

From equation (4.23), the time required to attain λ_{tv}, τ^*, was obtained as follows:

$$\tau^* = \left[-\frac{1}{\beta} W_n \left(\frac{-2\lambda_{tv}\left[1-(1+\beta T)e^{-\beta T}\right]}{\beta \chi^2(2n;\gamma)} \right) \right] - T \tag{4.24}$$

Equation (4.24) is the first formula of Proposition II.

When $\boldsymbol{\beta}$ is unknown, the time required to attain the target value, λ_{tv}, is τ, which is the solution to:

$$\gamma = 1 - \frac{1}{k} \sum_{h=0}^{n-1} \int_0^{\infty} \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}} \lambda_{tv} \right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}} \lambda_{tv}} \frac{\beta^{2n-1} e^{-\beta \sum_{i=1}^{n} t_i}}{\left[1-(1+\beta T)e^{-\beta T}\right]^n} \, d\beta \tag{4.25}$$

4.3.3 Proposition III

The probability that at most k failures will occur in the time interval $(T, \tau]$, $\tau > T$ is given by:

$$\gamma_k = \begin{cases} \frac{\left[1-(1+\beta T)e^{-\beta T}\right]^n}{\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]^n} \sum_{j=n}^{n+k} \binom{j-n}{n-1} \frac{\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]^j}{\left[1-(1+\beta \tau)e^{-\beta \tau}\right]^j} & \textit{if } \beta \textit{ is known} \\ \sum_{j=n}^{n+k} \frac{\Gamma(j)}{c(j-n)!\Gamma(n)} \int_0^{\infty} \beta^{2n-1} e^{-\beta \sum_{i=1}^{n} t_i} \frac{\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]^{j-n}}{\left[1-(1+\beta \tau)e^{-\beta \tau}\right]^j} d\beta & \textit{if } \beta \textit{ is unknown} \end{cases}$$

Proof:

The probability is given by:

$$\gamma_k = \Pr[\, N(\tau) \leq n + k \mid Y_{obs} \,]$$

When $\boldsymbol{\beta}$ is known:

$$\gamma_k = \int_0^{\infty} \Pr[\, N(\tau) \leq n + k \mid Y_{obs} \,] \, \pi(\, \alpha \mid Y_{obs} \,) d\alpha \tag{4.26}$$

But $\pi(\, \alpha \mid Y_{obs} \,)$ is given by equation (4.6) and

$\Pr[\,N(\tau) \leq n+k \mid Y_{obs}\,] = \sum_{j=n}^{n+k} f(Y_{obs}, N(\tau)=j \mid \alpha) \,/\, f(\,Y_{obs} \mid \alpha\,)$ (4.27)

From equation (4.1), $f(\,Y_{obs} \mid \alpha\,) = \alpha^n \beta^{2n} (\prod_{i=1}^{n} t_i) e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}$ and

$f(\,Y_{obs}, N(\tau)=j \mid \alpha\,) = \int_{D(j-n:T,\tau)} f(Y_{obs}, t_{n+1}, \ldots, t_j, N(\tau)=j) \prod_{l=n+1}^{j} dt_l$

$= \int_{D(j-n:T,\tau)} \alpha^j \beta^{2j} \left(\prod_{i=1}^{j} t_i\right) e^{-\beta \sum_{i=1}^{j} t_i} e^{-\alpha[1-(1+\beta T)e^{-\beta T}]} \prod_{l=n+1}^{j} dt_l$

$= \alpha^j \beta^{2j} (\prod_{i=1}^{n} t_i) e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha[1-(1+\beta\tau)e^{-\beta\tau}]} \int_{D(j-n:T,\tau)} \left(\prod_{l=n+k}^{j} t_l\right) e^{-\beta \sum_{l=n+k}^{j} t_l} \prod_{l=n+1}^{j} dt_l$ (4.28)

The integral part was solved as follows:

$\int_0^t t e^{-\beta t} dt = \frac{1}{\beta^2}\left[1-(1+\beta t)e^{-\beta t}\right]$

Substituting the limits T and τ; $\frac{1}{\beta^2}\left[1-(1+\beta\tau)e^{-\beta\tau}\right] - \frac{1}{\beta^2}\left[1-(1+\beta T)e^{-\beta T}\right]$, which reduces to $\frac{1}{\beta^2}\left[\left(\beta T e^{-\beta T} + e^{-\beta T}\right) - \left(\beta\tau e^{-\beta\tau} + e^{-\beta\tau}\right)\right]$. Therefore, the integral part of equation (4.28) was obtained as;

$\int_{D(j-n:T,\tau)} \left(\prod_{l=n+k}^{j} t_l\right) e^{-\beta \sum_{l=n+k}^{j} t_l} \prod_{l=n+1}^{j} dt_l = \frac{1}{\beta^{2(j-n)}} \frac{\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta\tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]^{j-n}}{(j-n)!}$ (4.29)

Substituting equation (4.29) into equation (4.28) yielded:

$$f(\,Y_{obs}, N(\tau)=j \mid \alpha\,) =$$
$$\alpha^j \beta^{2j} (\prod_{i=1}^{n} t_i) e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha[1-(1+\beta\tau)e^{-\beta\tau}]} \frac{1}{\beta^{2(j-n)}} \frac{\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta\tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]^{j-n}}{(j-n)!}$$

From equation (4.27):

$$f(\,Y_{obs}, N(\tau)=j \mid \alpha\,)/f(\,Y_{obs} \mid \alpha\,) =$$
$$\frac{\alpha^j \beta^{2j} (\prod_{i=1}^{n} t_i) e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha\left[1-(1+\beta\tau)e^{-\beta\tau}\right]} \frac{1}{\beta^{2(j-n)}} \frac{\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta\tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]^{j-n}}{(j-n)!}}{\alpha^n \beta^{2n} (\prod_{i=1}^{n} t_i) e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]}}$$

which reduces to:

$$f(\,Y_{obs}, N(\tau)=j \mid \alpha\,)/f(\,Y_{obs} \mid \alpha\,) =$$
$$\alpha^{j-n} e^{-\alpha[(\beta T e^{-\beta T}+e^{-\beta T})-(\beta\tau e^{-\beta\tau}+e^{-\beta\tau})]} \frac{\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta\tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]^{j-n}}{(j-n)!}$$

Thus, equation (4.27) becomes:

$$\Pr[\, N(\tau) \le n + k \mid Y_{obs}\,] = \sum_{j=n}^{n+k} f(Y_{obs}, N(\tau) = j \mid \alpha) \,/\, f(\, Y_{obs} \mid \alpha\,)$$

$$= \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta\tau e^{-\beta\tau} + e^{-\beta\tau})]} \frac{[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta\tau e^{-\beta\tau} + e^{-\beta\tau})]^{j-n}}{(j-n)!} \quad (4.30)$$

Hence, equation (4.26) becomes:

$$\gamma_k = \int_0^\infty \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta\tau e^{-\beta\tau} + e^{-\beta\tau})]} \frac{[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta\tau e^{-\beta\tau} + e^{-\beta\tau})]^{j-n}}{(j-n)!} \left\{ [\Gamma(n)]^{-1} \alpha^{n-1} e^{-\alpha[1-(1+\beta T)e^{-\beta T}]} \left[1 - (1+\beta T)e^{-\beta T}\right]^n \right\} d\alpha$$

$$= \sum_{j=n}^{n+k} \frac{[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta\tau e^{-\beta\tau} + e^{-\beta\tau})]^{j-n} \left[1-(1+\beta T)e^{-\beta T}\right]^n}{(j-n)!\,\Gamma(n)} \int_0^\infty \alpha^{j-n} e^{-\alpha[1-(1+\beta\tau)e^{-\beta\tau}]} d\alpha$$

$$= \sum_{j=n}^{n+k} \frac{[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta\tau e^{-\beta\tau} + e^{-\beta\tau})]^{j-n} \left[1-(1+\beta T)e^{-\beta T}\right]^n}{(j-n)!\,\Gamma(n)} \frac{\Gamma(j)}{\left[1-(1+\beta\tau)e^{-\beta\tau}\right]^j} \int_0^\infty \frac{\left[1-(1+\beta\tau)e^{-\beta\tau}\right]^j}{\Gamma(j)} \alpha^{j-n} e^{-\alpha[1-(1+\beta\tau)e^{-\beta\tau}]} d\alpha \quad (4.31)$$

The integral part of the equation (4.31) is a gamma distribution with parameters j and $\left[1-(1+\beta\tau)e^{-\beta\tau}\right]$, and thus integrates to 1. Hence, equation (4.31) reduces to:

$$\gamma_k = \sum_{j=n}^{n+k} \frac{[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta\tau e^{-\beta\tau} + e^{-\beta\tau})]^{j-n} \left[1-(1+\beta T)e^{-\beta T}\right]^n \Gamma(j)}{(j-n)!\,\Gamma(n) \left[1-(1+\beta\tau)e^{-\beta\tau}\right]^j} \quad (4.32)$$

Equation (4.32) was rearranged to obtain:

$$\gamma_k = \frac{\left[1-(1+\beta T)e^{-\beta T}\right]^n}{[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta\tau e^{-\beta\tau} + e^{-\beta\tau})]^n} \sum_{j=n}^{n+k} \binom{j-1}{n-1} \frac{[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta\tau e^{-\beta\tau} + e^{-\beta\tau})]^j}{\left[1-(1+\beta\tau)e^{-\beta\tau}\right]^j} \quad (4.33)$$

Equation (4.33) is the first formula of proposition III.

When $\boldsymbol{\beta}$ is unknown:

$$\gamma_k = \int_0^\infty \int_0^\infty \Pr[\, N(\tau) \le n + k \mid Y_{obs}\,]\, P(\,\alpha, \beta \mid Y_{obs}\,) d\alpha d\beta$$

where $\Pr[\, N(\tau) \le n + k \mid Y_{obs}\,]$ and $\pi(\,\alpha, \beta \mid Y_{obs}\,)$ are given by equations (4.30) and (4.12) respectively.

$$\gamma_k = \sum_{j=n}^{n+k} \frac{1}{c(j-n)!\Gamma(n)} \iint_0^\infty \alpha^{j-n} e^{-\alpha[(\beta T e^{-\beta T}+e^{-\beta T})-(\beta\tau e^{-\beta\tau}+e^{-\beta\tau})]} \left[\left(\beta T e^{-\beta T} + e^{-\beta}\right) - \left(\beta\tau e^{-\beta\tau} + e^{-\beta\tau}\right)\right]^{j-n} \alpha^{n-1}\beta^{2n-1} e^{-\beta\sum_{i=1}^{n} t_i} e^{-\alpha[1-(1+\beta T)e^{-\beta T}]} d\alpha d\beta$$

$$= \sum_{j=n}^{n+k} \frac{\Gamma(j)}{c(j-n)!\Gamma(n)} \int_0^\infty \beta^{2n-1} e^{-\beta\sum_{i=1}^{n} t_i} \frac{\left[(\beta T e^{-\beta T}+e^{-\beta T})-(\beta\tau e^{-\beta\tau}+e^{-\beta\tau})\right]^{j-n}}{\left[1-(1+\beta\tau)e^{-\beta\tau}\right]^{j}} d\beta \qquad (4.34)$$

where $c = k$ as used in equation (4.12). Letter c has been substituted for k because the summation in equation (4.34) is from n to $(n+k)$, and the k*'s* are not the same. Equation (4.34) implies the second formula in proposition III.

4.3.4 Proposition IV

The Bayesian UPL of $\lambda(t) = \alpha\beta^2 t e^{-\beta t}$ with level, γ, was obtained as:

$$\lambda_U^{(\beta)} = \begin{cases} \frac{(\beta^2 t e^{-\beta t})\chi^2(2n;\gamma)}{2[1-(1+\beta T)e^{-\beta T}]} & if\ \beta\ is\ known \\ \lambda_{tv} & if\ \beta\ is\ unknown \end{cases}$$

Remark 2:

The second part of Proposition IV is such that λ_{tv} is the solution to the equation;

$$\gamma = 1 - \frac{1}{k}\sum_{h=0}^{n-1} \int_0^\infty \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\lambda_{tv}} \frac{\beta^{2n-1}e^{-\beta\sum_{i=1}^{n} t_i}}{\left[1-(1+\beta T)e^{-\beta T}\right]^n} d\beta$$

Proof:

When $\boldsymbol{\beta}$ is known, given a predetermined τ $(\tau > T)$, the Bayesian UPL for $\lambda(t)$ with level γ, denoted by $\lambda_U^{(\beta)}$ satisfies $\gamma = \Pr\left(\lambda_t \le \lambda_U^{(\beta)}(\tau) \mid Y_{obs}\right)$. From equations (4.15) and (4.21):

$$\gamma = \int_0^{\lambda_U^{(\beta)}(\tau)} f \left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 t e^{-\beta t}}\right]^n \frac{\lambda_\tau^{n-1}}{\Gamma(n)} e^{-\lambda_\tau\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 t e^{-\beta t}}\right]} d\lambda_\tau \qquad (4.35)$$

From equation (4.27):

$$2\left[\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 t e^{-\beta t}}\right]\lambda_U^{(\beta)}(\tau) = \chi^2(2n;\gamma) \qquad (4.36)$$

Making $\lambda_U^{(\beta)}(\tau)$ in equation (4.36) the subject, we get;

$$\lambda_U^{(\beta)}(\tau) = \frac{(\beta^2 te^{-\beta t})\chi^2(2n;\gamma)}{2[1-(1+\beta T)e^{-\beta T}]} \tag{4.37}$$

Equation (4.37) implies the first part of proposition IV.

When $\boldsymbol{\beta}$ is unknown, the Bayesian UPL for $\lambda(t) = \alpha\beta^2 te^{-\beta t}$ with level γ is λ_{tv}, which is the solution to;

$$\gamma = 1 - \frac{1}{k}\sum_{h=0}^{n-1}\int_0^\infty \frac{\left(\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 te^{-\beta t}}\lambda_{tv}\right)^h}{h!} e^{-\frac{1-(1+\beta T)e^{-\beta T}}{\beta^2 te^{-\beta t}}\lambda_{tv}} \frac{\beta^{2n-1}e^{-\beta\sum_{i=1}^n t_i}}{[1-(1+\beta T)e^{-\beta T}]^n}\, d\beta \tag{4.38}$$

4.4 Main Results for Single-Sample Prediction Using Informative Priors

In this section, formulas to address the proposed issues were derived, assuming that α and β are gamma-distributed. The joint density given in equation (4.1) was used.

Case 1: When β is known, an informative prior was adopted, assuming that $\alpha \sim Gamma(a,b)$, where *a* and *b* are known hyperparameters.

$$\pi(\alpha) \propto \alpha^{a-1}e^{-b\alpha} \tag{4.39}$$

The posterior distribution of α obtained from equation (4.2) is as follows:

$$\pi(\alpha \mid Y_{obs}) = \frac{f(Y_{obs}|\alpha,\beta)\pi(\alpha,\beta)}{\int_0^\infty\int_0^\infty f(Y_{obs}|\alpha,\beta)\pi(\alpha,\beta)d\alpha d\beta}$$

Equations (4.1) and (4.39) were used to obtain:

$$\pi(\alpha \mid Y_{obs}) = \frac{\alpha^{n+a-1}\beta^{2n}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}e^{-\alpha[b-1-(1+\beta T)e^{-\beta T}]}}{\int_0^\infty \alpha^{n+a-1}\beta^{2n}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}e^{-\alpha[b-1-(1+\beta T)e^{-\beta T}]}d\alpha}$$

$$= \frac{\beta^{2n}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}\,\alpha^{n+a-1}e^{-\alpha[b-1-(1+\beta T)e^{-\beta T}]}}{\beta^{2n}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}\int_0^\infty \alpha^{n+a-1}e^{-\alpha[b-1-(1+\beta T)e^{-\beta T}]}d\alpha} \tag{4.40}$$

The denominator of equation (4.40) can be written as:

$$\beta^{2n}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}\frac{\Gamma(n+a)}{[b-1-(1+\beta T)e^{-\beta T}]^{n+a}}\int_0^\infty \frac{[b-1-(1+\beta T)e^{-\beta T}]^{n+a}}{\Gamma(n+a)}\,\alpha^{n+a-1}e^{-\alpha[b-1-(1+\beta T)e^{-\beta T}]}d\alpha$$

which reduces to:

$$\beta^{2n}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}\frac{\Gamma(n+a)}{[b-1-(1+\beta T)e^{-\beta T}]^{n+a}}\,.$$

Equation (4.40) becomes:

$$\pi(\alpha \mid Y_{obs}) = \frac{\beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}\, \alpha^{n+a-1}e^{-\alpha\left[b-1-(1+\beta T)e^{-\beta T}\right]}}{\beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}\frac{\Gamma(n+a)}{\left[b-1-(1+\beta T)e^{-\beta T}\right]^{n+a}}},$$

which reduces to:

$$\pi(\alpha \mid Y_{obs}) = [\Gamma(n+a)]^{-1}\, \alpha^{n+a-1}e^{-\alpha\left[b-1-(1+\beta T)e^{-\beta T}\right]}\left[b-1-(1+\beta T)e^{-\beta T}\right]^{n+a} \quad (4.41)$$

Case 2: When β is unknown, the study adopts a joint prior distribution of α and β, assuming they are independent and $\beta \sim Gamma\ (c, d)$, where *c* and *d* are known.

$\pi(\beta)\ \propto\ \beta^{c-1}e^{-d\beta}$

Since the parameters are independent, the joint prior function was obtained as follows:

$\pi(\alpha,\beta)\ \propto\ \pi(\alpha)\pi(\beta)$, implying that;

$$\pi(\alpha,\beta)\ \propto\ \alpha^{a-1}e^{-b\alpha}\,\beta^{c-1}e^{-d\beta} \quad (4.42)$$

The joint posterior density of α and β obtained from equation (4.2) and equation (4.42) is as follows:

$$\pi(\alpha,\beta \mid Y_{obs}) = \frac{f(Y_{obs}|\alpha,\beta)\pi(\alpha,\beta)}{\int_0^\infty\int_0^\infty f(Y_{obs}|\alpha,\beta)\pi(\alpha,\beta)d\alpha d\beta}$$

$$= \frac{\alpha^{a-1}e^{-b\alpha}\,\beta^{c-1}e^{-d\beta}\alpha^{n}\beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]}}{\iint_0^\infty \alpha^{a-1}e^{-b\alpha}\,\beta^{c-1}e^{-d\beta}\,\alpha^{n}\beta^{2n}(\prod_{i=1}^{n} t_i)e^{-\beta\sum_{i=1}^{n} t_i}e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]}d\alpha d\beta}$$

$$= \frac{\alpha^{n+a-1}\beta^{2n+c-1}(\prod_{i=1}^{n} t_i)e^{-\beta(d+\sum_{i=1}^{n} t_i)}\ e^{-\alpha\left[b+1-(1+\beta T)e^{-\beta T}\right]}}{\iint_0^\infty \alpha^{n+a-1}\beta^{2n+c-1}(\prod_{i=1}^{n} t_i)e^{-\beta(d+\sum_{i=1}^{n} t_i)}\ e^{-\alpha\left[b+1-(1+\beta T)e^{-\beta T}\right]}d\alpha d\beta}$$

$$= \frac{\alpha^{n+a-1}\beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^{n} t_i)}\ e^{-\alpha\left[b+1-(1+\beta T)e^{-\beta T}\right]}}{\left\{\left(\int_0^\infty \beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^{n} t_i)}d\beta\right)\left(\int_0^\infty \alpha^{n+a-1}e^{-\alpha\left[b+1-(1+\beta T)e^{-\beta T}\right]}d\alpha\right)\right\}}$$

$$= \frac{\alpha^{n+a-1}\beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^{n} t_i)}\ e^{-\alpha\left[b+1-(1+\beta T)e^{-\beta T}\right]}}{\Gamma(n+a)\int_0^\infty \frac{\beta^{2n+c-1}e^{-\beta\left(d+\sum_{i=1}^{n} t_i\right)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}}d\beta} \quad (4.43)$$

Let $q = \int_0^\infty \frac{\beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^{n} t_i)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}}d\beta.$

Equation (4.43) becomes:

$$\pi(\alpha,\beta \mid Y_{obs}) = [q\Gamma(n+a)]^{-1}\alpha^{n+a-1}\beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^{n} t_i)}\ e^{-\alpha\left[b+1-(1+\beta T)e^{-\beta T}\right]} \quad (4.44)$$

4.4.1 Proposition I.1

The probability that the target value λ_{tv} will be achieved at time τ $(\tau > T)$ is:

$$\gamma = \begin{cases} 1-\sum_{h=0}^{n+a-1}\frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!}e^{-\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\lambda_{tv}\right]} & if\ \beta\ is\ known \\ 1-\frac{1}{q}\sum_{h=0}^{n+a-1}\int_0^\infty\frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!}e^{-\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\lambda_{tv}\right]}\frac{\beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^n t_i)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}}\,d\beta & if\ \beta\ is\ unknown \end{cases}$$

Proof:

Let $\pi(Y_{obs} \mid \lambda_\tau)$ denote the posterior density of $\lambda(t) = \alpha\beta^2 te^{-\beta t}$. Thus, the probability that λ_{tv} will be achieved at time τ $(\tau > T)$ is given by (4.15). **When $\boldsymbol{\beta}$ is known**, $\alpha = \frac{\lambda(t)}{\beta^2 te^{-\beta t}}$, as obtained from the intensity function, and $\frac{d\alpha}{d\lambda_\tau} = \frac{1}{\beta^2 te^{-\beta t}}$. The posterior distribution of λ_τ is $\pi(\lambda_\tau \mid Y_{obs}) = \pi(\alpha \mid Y_{obs})\left|\frac{d\alpha}{d\lambda_\tau}\right|$. Thus, the posterior density becomes:

$$\pi(\lambda_\tau \mid Y_{obs}) = \frac{1}{[\Gamma(n+a)]}\left(\frac{\lambda_\tau}{\beta^2\tau e^{-\beta\tau}}\right)^{n+a-1} e^{-\left(\frac{\lambda_\tau}{\beta^2\tau e^{-\beta\tau}}\right)\left[b+1-(1+\beta T)e^{-\beta T}\right]}\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}.\frac{1}{\beta^2\tau e^{-\beta\tau}},$$

which reduces to:

$$\pi(\lambda_\tau \mid Y_{obs}) = \frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\right)^{n+a}}{\Gamma(n+a)}\lambda_\tau^{n+a-1}e^{-\lambda_\tau\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\right)} \qquad (4.45)$$

From equation (4.45), λ_τ follows a gamma distribution with parameters $(n+a)$ and $\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\right)$. Recall the relationship between gamma and Poisson distributions given by equation (4.17) as $\frac{\beta^\alpha}{\Gamma(n)}\int_0^\lambda x^{\alpha-1}e^{-\beta x}dx = 1-\sum_{h=0}^{\alpha-1}\frac{(\beta\lambda)^h}{h!}e^{-\beta\lambda}$.

From equations (4.15) and (4.45):

$$\gamma = 1-\sum_{h=0}^{n+a-1}\frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!}e^{-\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\right]\lambda_{tv}} \qquad (4.46)$$

Equation (4.46) is the same as the first formula in Proposition I.1.

When $\boldsymbol{\beta}$ is unknown: From the intensity function, $\alpha = \frac{\lambda_\tau}{\beta^2 \tau e^{-\beta\tau}}$, and let $\beta = \beta$. The Jacobian is $\frac{d(\alpha,\beta)}{d(\lambda_\tau,\beta)} = \frac{1}{\beta^2 \tau e^{-\beta\tau}}$. The joint posterior density of (λ_τ, β) is given by;

$$\pi(\lambda_\tau, \beta \mid Y_{obs}) = \pi(\alpha, \beta \mid Y_{obs}) \left|\frac{d(\alpha,\beta)}{d(\lambda_\tau,\beta)}\right|$$

From equation (4.44):

$$\pi(\lambda_\tau, \beta \mid Y_{obs}) = [q\Gamma(n+a)]^{-1} \alpha^{n+a-1} \beta^{2n+c-1} e^{-\beta(d+\sum_{i=1}^{n} t_i)}\, e^{-\alpha[b+1-(1+\beta T)e^{-\beta T}]} . \frac{1}{\beta^2 t e^{-\beta t}}$$

$$= \left(\frac{\lambda_\tau}{\beta^2 \tau e^{-\beta\tau}}\right)^{n+a-1} \frac{\beta^{2n+c-1}}{q\Gamma(n+a)} e^{-\beta(d+\sum_{i=1}^{n} t_i)} e^{-\lambda_\tau\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]} . \frac{1}{\beta^2 \tau e^{-\beta\tau}}$$

$$= \left(\frac{1}{\beta^2 \tau e^{-\beta\tau}}\right)^{n+a-1} \frac{\beta^{2n+c-1}}{q\Gamma(n+a)} e^{-\beta(d+\sum_{i=1}^{n} t_i)}\ \lambda_\tau^{n+a-1} e^{-\lambda_\tau\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]} . \frac{1}{\beta^2 \tau e^{-\beta\tau}}$$

$$= \left(\beta^2 \tau e^{-\beta\tau}\right)^{-(n+a)} \frac{\beta^{2n+c-1}}{q\Gamma(n+a)} e^{-\beta(d+\sum_{i=1}^{n} t_i)}\ \lambda_\tau^{n+a-1} e^{-\lambda_\tau\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]}$$

$$=$$

$$\frac{\left(\beta^2 t e^{-\beta t}\right)^{-(n+a)}}{q\Gamma(n+a)} \beta^{2n+c-1} e^{-\beta(d+\sum_{i=1}^{n} t_i)} \frac{\Gamma(n+a)}{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right)^{n+a}} \frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 t e^{-\beta t}}\right)^{n+a}}{\Gamma(n+a)} \lambda_\tau^{n+a-1} e^{-\lambda_\tau\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]}$$

$$= \frac{\beta^{2n+c-1} e^{-\beta(d+\sum_{i=1}^{n} t_i)}}{q\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}} \frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right)^{n+a}}{\Gamma(n+a)}\ \lambda_\tau^{n+a-1} e^{-\lambda_\tau\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]} \qquad (4.47)$$

From equation (4.15), equation (4.17), and equation (4.47):

$$\gamma = \frac{1}{q}\int_0^\infty \frac{\beta^{2n+c-1} e^{-\beta(d+\sum_{i=1}^{n} t_i)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}} \left\{1 - \sum_{h=0}^{n+a-1} \frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\lambda_{tv}\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]}\right\} d\beta$$

$$= \frac{1}{q}\int_0^\infty \frac{\beta^{2n+c-1} e^{-\beta(d+\sum_{i=1}^{n} t_i)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}} d\beta\, -$$

$$\frac{1}{q}\sum_{h=0}^{n+a-1} \int_0^\infty \frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\lambda_{tv}\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]} \frac{\beta^{2n+c-1} e^{-\beta(d+\sum_{i=1}^{n} t_i)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}} d\beta$$

$$= 1 - \frac{1}{q}\sum_{h=0}^{n+a-1}\int_0^\infty \frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\lambda_{tv}\right)^h}{h!} e^{-\lambda_{tv}\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]} \frac{\beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^n t_i)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}} d\beta \quad (4.48)$$

Equation (4.48) is the same as the equation in the second formula of proposition I.1.

4.4.2 Proposition II.1

The time τ^* required to attain λ_{tv} for a given level γ is;

$$\tau^* = \begin{cases} \left[-\frac{1}{\beta}W_n\left(\frac{-2\lambda_{tv}\left[b+1-(1+\beta T)e^{-\beta T}\right]}{\beta\chi^2(2n;\gamma)}\right)\right] - T & \text{if } \beta \text{ is known} \\ \tau - T & \text{if } \beta \text{ is unknown} \end{cases}$$

Proof:

The time required to attain λ_{tv} for a given level γ is obtained as $\tau^* = \tau - T$, where τ satisfies equation (4.15). When β is known, from equation (4.45), $2\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right)\lambda_\tau$ follows a Chi-square distribution with *2n* degrees of freedom. Thus;

$$2\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right)\lambda_{tv} = \chi^2(2n;\gamma) \quad (4.49)$$

$$\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}} = \frac{\chi^2(2n;\gamma)}{2\lambda_{tv}} \quad (4.50)$$

$$\tau e^{-\beta\tau} = \frac{2\lambda_{tv}\left[b+1-(1+\beta T)e^{-\beta T}\right]}{\beta^2\chi^2(2n;\gamma)} \quad (4.51)$$

Making τ the subject in equation (4.51), we have;

$$\tau = -\frac{1}{\beta}W_n\left(\frac{-2\lambda_{tv}\left[b+1-(1+\beta T)e^{-\beta T}\right]}{\beta\chi^2(2n;\gamma)}\right), \text{ for } n \in \mathrm{Z} \quad (4.52)$$

From equation (4.52), we obtain τ^* as follows;

$$\tau^* = -\frac{1}{\beta}W_n\left(\frac{-2\lambda_{tv}\left[b+1-(1+\beta T)e^{-\beta T}\right]}{\beta\chi^2(2n;\gamma)}\right) - T \quad (4.53)$$

Equation (4.53) is the first formula in proposition II.1

When $\boldsymbol{\beta}$ is unknown, $\tau^* = \tau - T$, where τ is the solution to;

$$\gamma = 1 - \frac{1}{q}\sum_{h=0}^{n+a-1}\int_0^\infty \frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 t e^{-\beta t}}\lambda_{tv}\right)^h}{h!} e^{-\lambda_{tv}\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 t e^{-\beta t}}\right]} \frac{\beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^n t_i)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}} d\beta \quad (4.54)$$

4.4.3 Proposition III.1

The probability that at most k failures will occur in the time interval $(T, \tau]$, $\tau > T$ is given by:

$$\gamma_k = \begin{cases} \left[\frac{b+1-(1+\beta T)e^{-\beta T}}{(\beta T e^{-\beta T}+e^{-\beta T})-(\beta\tau e^{-\beta\tau}+e^{-\beta\tau})}\right]^n \left[\frac{b+1-(1+\beta T)e^{-\beta T}}{b+1-(1+\beta\tau)e^{-\beta\tau}}\right]^a \sum_{j=1}^{n+k}\binom{j+a-1}{n+a-1} \left[\frac{(\beta T e^{-\beta T}+e^{-\beta T})-(\beta\tau e^{-\beta\tau}+e^{-\beta\tau})}{b+1-(1+\beta\tau)e^{-\beta\tau}}\right]^j, \beta\ know \\ \sum_{j=n}^{n+k}\frac{\Gamma(j+a)}{q(j-n)!\Gamma(n+a)}\int_0^\infty \beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^n t_i)}\frac{[(\beta T e^{-\beta T}+e^{-\beta T})-(\beta\tau e^{-\beta\tau}+e^{-\beta\tau})]^{j-n}}{[b+1-(1+\beta T)e^{-\beta T}]^{j+a}}d\beta \quad if\ \beta\ is\ unknown \end{cases}$$

Proof:

The probability is given by:

$$\gamma_k = \Pr[\, N(\tau) \le n+k \mid Y_{obs}\,]$$

When β is known:

$$\gamma_k = \int_0^\infty \Pr[\, N(\tau) \le n+k \mid Y_{obs}, \alpha\,]\, P(\,\alpha \mid Y_{obs}\,)d\alpha \qquad (4.55)$$

where $p(\,\alpha \mid Y_{obs}\,)$ is given by equation and

$$\Pr[\, N(\tau) \le n+k \mid Y_{obs}, \alpha\,] = \sum_{j=n}^{n+k} f(Y_{obs}, N(\tau)=j \mid \alpha)\,/\,f(\,Y_{obs} \mid \alpha\,) \qquad (4.56)$$

From equation (4.1), $f(\,Y_{obs} \mid \alpha\,) = \alpha^n\beta^{2n}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}e^{-\alpha[1-(1+\beta T)e^{-\beta T}]}$ and

$$f(\,Y_{obs}, N(\tau)=j \mid \alpha\,) = \int_{D(j-n:T,\tau)} f(Y_{obs}, t_{n+1}, \dots, t_j, N(\tau)=j)\prod_{l=n+1}^{j} dt_l$$

$$= \int_{D(j-n:T,\tau)} \alpha^j\beta^{2j}\left(\prod_{i=1}^j t_i\right)e^{-\beta\sum_{i=1}^n t_i}e^{-\alpha\left[1-(1+\beta\tau)e^{-\beta\tau}\right]}\prod_{l=n+1}^{j} dt_l$$

$$= \alpha^j\beta^{2j}(\prod_{i=1}^n t_i)e^{-\beta\sum_{i=1}^n t_i}e^{-\alpha[1-(1+\beta\tau)e^{-\beta\tau}]}\int_{D(j-n:T,\tau)}\left(\prod_{l=n+k}^{j} t_l\right)e^{-\beta\sum_{l=n+k}^{j} t_l}\prod_{l=n+1}^{j} dt_l \quad (4.57)$$

The integral part of equation (4.57) is solved as follows;

$$\int_0^t te^{-\beta t}dt = \frac{1}{\beta^2}\left[1-(1+\beta t)e^{-\beta t}\right]$$

Substituting the limits T and τ, we get; $\frac{1}{\beta^2}\left[1-(1+\beta\tau)e^{-\beta\tau}\right] - \frac{1}{\beta^2}\left[1-(1+\beta T)e^{-\beta T}\right]$, which reduces to $\frac{1}{\beta^2}\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta\tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]$. Therefore, the integral part of equation (4.57) is obtained as:

$$\int_{D(j-n:T,\tau)} \left(\prod_{l=n+k}^{j} t_l\right) e^{-\beta \sum_{l=n+k}^{j} t_l} \prod_{l=n+1}^{j} dt_l = \frac{1}{\beta^{2(j-n)}} \frac{\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]^{j-n}}{(j-n)!} \quad (4.58)$$

Substituting equation (4.58) into equation (4.57), we obtain;

$$f(Y_{obs}, N(\tau) = j \mid \alpha) =$$

$$\alpha^{j} \beta^{2j} \left(\prod_{i=1}^{n} t_i\right) e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha\left[1-(1+\beta\tau)e^{-\beta\tau}\right]} \frac{1}{\beta^{2(j-n)}} \frac{\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]^{j-n}}{(j-n)!}$$

From equation (4.56):

$$f(Y_{obs}, N(\tau) = j \mid \alpha) / f(Y_{obs} \mid \alpha) =$$

$$\frac{\alpha^{j} \beta^{2j} \left(\prod_{i=1}^{n} t_i\right) e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha\left[1-(1+\beta\tau)e^{-\beta\tau}\right]} \frac{1}{\beta^{2(j-n)}} \frac{\left[\left(\beta T e^{-\beta T} + e^{-\beta T}\right) - \left(\beta \tau e^{-\beta \tau} + e^{-\beta \tau}\right)\right]^{j-n}}{(j-n)!}}{\alpha^{n} \beta^{2n} \left(\prod_{i=1}^{n} t_i\right) e^{-\beta \sum_{i=1}^{n} t_i} e^{-\alpha\left[1-(1+\beta T)e^{-\beta T}\right]}}$$

which reduces to:

$$f(Y_{obs}, N(\tau) = j \mid \alpha) / f(Y_{obs} \mid \alpha) =$$

$$\alpha^{j-n} e^{-\alpha\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]} \frac{\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]^{j-n}}{(j-n)!}$$

Thus, equation (4.56) becomes:

$$\Pr[N(\tau) \le n + k \mid Y_{obs}, \alpha] = \sum_{j=n}^{n+k} f(Y_{obs}, N(\tau) = j \mid \alpha) \,/\, f(Y_{obs} \mid \alpha)$$

$$= \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]} \frac{\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]^{j-n}}{(j-n)!} \quad (4.59)$$

Equation (4.55) becomes;

$$\gamma_k =$$

$$\int_0^{\infty} \sum_{j=n}^{n+k} \alpha^{j-n} e^{-\alpha\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]} \frac{\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]^{j-n}}{(j-n)!\,\Gamma(n+a)} \left\{\alpha^{n+a-1} e^{-\alpha\left[b+1-(1+\beta T)e^{-\beta T}\right]} \left[b + 1 - (1 + \beta T)e^{-\beta T}\right]^{n+a}\right\} d\alpha$$

$$=$$

$$\sum_{j=n}^{n+k} \frac{\left[(\beta T e^{-\beta T} + e^{-\beta T}) - (\beta \tau e^{-\beta \tau} + e^{-\beta \tau})\right]^{j-n} \left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}}{(j-n)!\,\Gamma(n+a)} \int_0^{\infty} \alpha^{j+a-1} e^{-\alpha\left[b+1-(1+\beta\tau)e^{-\beta\tau}\right]} d\alpha$$

$$= \sum_{j=n}^{n+k} \frac{\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta \tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]^{j-n}\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}}{(j-n)!\,\Gamma(n+a)} \frac{\Gamma(j+a)}{\left[b+1-(1+\beta\tau)e^{-\beta\tau}\right]^{j+a}} \int_0^\infty \frac{\left[b+1-(1+\beta\tau)e^{-\beta\tau}\right]^{j+a}}{\Gamma(j+a)} \alpha^{j+a-1} e^{-\alpha\left[b+1-(1+\beta\tau)e^{-\beta\tau}\right]} d\alpha \quad (4.60)$$

The integral part of equation (4.60) is a gamma distribution with parameters $(j+a)$ and $\left[b+1-(1+\beta\tau)e^{-\beta\tau}\right]$, thus integrates to 1. The equation becomes;

$$= \sum_{j=n}^{n+k} \frac{\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta \tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]^{j-n}\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}}{(j-n)!\,\Gamma(n+a)} \frac{\Gamma(j+a)}{\left[b+1-(1+\beta\tau)e^{-\beta\tau}\right]^{j+a}} \quad (4.61)$$

Equation (4.61) is rearranged to obtain;

$$\gamma_k = \left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta \tau e^{-\beta\tau}+e^{-\beta\tau}\right)}\right]^n \left[\frac{b+1-(1+\beta T)e^{-\beta T}}{b+1-(1+\beta\tau)e^{-\beta\tau}}\right]^a \sum_{j=1}^{n+1}\binom{j+a-1}{n+a-1}\left[\frac{\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta \tau e^{-\beta\tau}+e^{-\beta\tau}\right)}{b+1-(1+\beta\tau)e^{-\beta\tau}}\right]^j \quad (4.62)$$

Equation (4.62) is the first formula in proposition III.1.

When $\boldsymbol{\beta}$ is unknown, given that $\Pr[\,N(\tau) \le n+k \mid Y_{obs}, \alpha, \beta\,]$ and $\pi(\,\alpha, \beta \mid Y_{obs}\,)$ are presented in equation (4.59) and equation (4.44):

$$\gamma_k = \int_0^\infty \int_0^\infty \Pr[\,N(\tau) \le n+k \mid Y_{obs}\,]\, \pi(\,\alpha,\beta \mid Y_{obs}\,) d\alpha d\beta$$

$$= \sum_{j=n}^{n+k} \frac{1}{q(j-n)!\Gamma(n+a)} \iint_0^\infty \alpha^{j-n} e^{-\alpha\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta \tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]}\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta \tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]^{j-n} \alpha^{n+a-1}\beta^{2n+c-1}e^{-\beta\left(d+\sum_{i=1}^n t_i\right)}\, e^{-\alpha\left[b+1-(1+\beta T)e^{-\beta T}\right]} d\alpha d\beta$$

$$= \sum_{j=n}^{n+k} \frac{\Gamma(j+a)}{q(j-n)!\Gamma(n+a)} \int_0^\infty \beta^{2n+c-1} e^{-\beta\left(d+\sum_{i=1}^n t_i\right)} \frac{\left[\left(\beta T e^{-\beta T}+e^{-\beta T}\right)-\left(\beta \tau e^{-\beta\tau}+e^{-\beta\tau}\right)\right]^{j-n}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{j+a}} d\beta \quad (4.63)$$

Equation (4.63) is the second formula in proposition III.1.

4.4.4 Proposition IV.1

For a given level γ, the Bayesian UPL of $\lambda_\tau = \alpha\beta^2 t e^{-\beta t}$ is obtained as;

$$\lambda_U^{(\beta)} = \begin{cases} \dfrac{\left(\beta^2 t e^{-\beta t}\right)\chi^2(2n;\gamma)}{2\left[b+1-(1+\beta T)e^{-\beta T}\right]} & \text{if } \beta \text{ is known} \\ \lambda_{tv} & \text{if } \beta \text{ is unknown} \end{cases}$$

Remark 3:

The second part o8f proposition IV.1 is such that λ_{tv} is the solution to the equation;

$$\gamma = 1 - \frac{1}{q}\sum_{h=0}^{n+a-1}\int_0^{\infty} \frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 te^{-\beta t}}\lambda_{tv}\right)^h}{h!} e^{-\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 te^{-\beta t}}\right)\lambda_{tv}} \frac{\beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^{n}t_i)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}}\, d\beta$$

Proof:

When $\boldsymbol{\beta}$ is known, given a predetermined τ $(\tau > T)$, the Bayesian UPL for λ_τ with level γ, denoted by $\lambda_U^{(\beta)}$ satisfies $\gamma = \Pr\left(\lambda_t \le \lambda_U^{(\beta)}(\tau) \mid Y_{obs}\right)$. From equations (4.15) and (4.49),

$$\gamma = \int_0^{\lambda_U^{(\beta)}(\tau)} f \left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta\tau}}\right]^{n+a} \frac{\lambda_\tau^{n+a-1}}{\Gamma(n+a)}\; e^{-\lambda_\tau\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\right]} d\lambda_\tau \tag{4.64}$$

This implies that,

$$2\left[\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2\tau e^{-\beta\tau}}\right]\lambda_U^{(\beta)}(\tau) = \chi^2(2n;\gamma) \tag{4.65}$$

Making $\lambda_U^{(\beta)}(\tau)$ in equation (4.65) the subject, we get;

$$\lambda_U^{(\beta)}(\tau) = \frac{(\beta^2\tau e^{-\beta\tau})\chi^2(2n;\gamma)}{2\left[b+1-(1+\beta T)e^{-\beta T}\right]} \tag{4.66}$$

Equation (4.66) implies the first part of proposition IV.1.

When $\boldsymbol{\beta}$ is unknown, the Bayesian UPL for $\lambda_\tau = \alpha\beta^2 te^{-\beta t}$ with level γ is λ_{tv}, which is the solution to:

$$\gamma = 1 - \frac{1}{q}\sum_{h=0}^{n+a-1}\int_0^{\infty} \frac{\left(\frac{b+1-(1+\beta T)e^{-\beta T}}{\beta^2 \tau e^{-\beta \tau}}\lambda_{tv}\right)^h}{h!} e^{-\frac{\left(b+1-(1+\beta T)e^{-\beta T}\right)}{\beta^2 \tau e^{-\beta \tau}}\lambda_{tv}} \frac{\beta^{2n+c-1}e^{-\beta(d+\sum_{i=1}^{n}t_i)}}{\left[b+1-(1+\beta T)e^{-\beta T}\right]^{n+a}}\, d\beta \tag{4.67}$$

4.5 Real Examples for Single-Sample Bayesian Prediction

4.5.1 Results Obtained Using Non-Informative Priors

In this section, secondary software failure data in Table 4.3, obtained from Ehrlich *et al.* (1993), was used for single-sample Bayesian prediction, and the results are presented and discussed.

Table 4.3: Time between failures data.

Index	Failure Time	Inter-failure time	Index	Failure Time	Inter-failure time
1	5.5	5.5	12	166.99	14.6
2	7.33	1.83	13	178.41	11.41
3	10.08	2.75	14	197.35	18.94
4	80.97	70.89	15	262.65	65.3
5	84.91	3.94	16	262.69	0.04
6	99.89	14.98	17	388.36	125.67
7	103.36	3.47	18	471.05	82.69
8	113.32	9.96	19	471.5	0.46
9	124.71	11.39	20	503.11	31.61
10	144.59	19.88	21	632.42	129.31
11	152.4	7.81	22	680.02	47.6

The study used failure times $0 < t_1 < t_2 < \cdots < t_{22}$, where n = 22. The data was used by Ehrlich *et al.* (1993) and Hung-Cuong and Quyet-Thang (2015), with the latter arguing that the data has been widely used in assessing software reliability models. For the case where β is assumed to be known, the MLE was computed, fixing parameter values at $\alpha = 18$ and $\beta = 0.00342$, where α and β were close to the Maximum Likelihood estimates obtained by Hung-Cuong and Quyet-Thang (2015). The MLE for β, for the Delayed S-shaped model, was obtained as $\beta = 0.007609807$. The study used this value for the cases where β was assumed to be known.

Proposition I: Suppose the target value is given by $\lambda_{tv} = 0.02$. This implies that the probability of software failure occurring in the future time period is 0.02. At time $T = 100h$, the MLE of the achieved software rate for this software is $\lambda = \alpha\beta e^{-100\beta} = 0.0640$, which is greater than 0.02. Thus, the target value cannot be achieved at the initial time window, $T = 100h$. Assuming a time greater than $100h$, $\tau = 500h$, and the focus is to determine the probability of achieving λ_{tv} at this time: (i) When β is known ($\beta = 0.007609807$), from the first formula in Proposition I, $\gamma = 9.0 \times 10^{-9}$. Therefore, it is almost unlikely that the target value will be achieved. (ii) When β is unknown, from the second formula in Proposition I, $\gamma =$

3.0×10^{-7}. It is unlikely that the target software failure intensity of 0.02 will be achieved at time $\tau = 500h$.

Since the target value was not achieved at $\tau = 500h$, it would be of interest to assess the relationship between the probability and time by varying τ, while holding $\lambda_{tv} = 0.02$ constant. This was illustrated in the case when β is unknown. Table 4.4 shows the probabilities and time, indicating that it is unlikely that the targeted failure intensity will be achieved between $130h$ and $530h$. However, it is almost certain that at $830h$ and above, the targeted software failure intensity will be achieved. It is crucial to note that these values do not indicate the exact time at which the failure intensity would be achieved but rather the time by which it is expected to have been achieved with some probability. Figure 4.7 displays the relationship, indicating that the probability of achieving a software failure intensity of 0.02 increases with time. It can be observed that the probability increases rapidly between $630h$ and $780h$, suggesting that between these time periods and/or above, the targeted software failure intensity is achievable.

Table 4.4: The probability that the software failure rate of 0.02 will be achieved at different periods, τ.

Time period, τ in h	$P(\lambda_{tv} = 0.02)$	**Time period, τ in h**	$P(\lambda_{tv} = 0.02)$
130	0	580	3.15×10^{-4}
180	0	630	1.165×10^{-2}
230	1×10^{-16}	680	1.634×10^{-1}
280	2×10^{-16}	730	6.803×10^{-1}
330	1.72×10^{-14}	780	9.824×10^{-1}
380	2.09×10^{-12}	830	9.9996×10^{-1}
430	2.9×10^{-9}	880	1.0×10^{0}
480	3.96×10^{-8}	930	1.0×10^{0}
530	4.36×10^{-6}	980	1.0×10^{0}

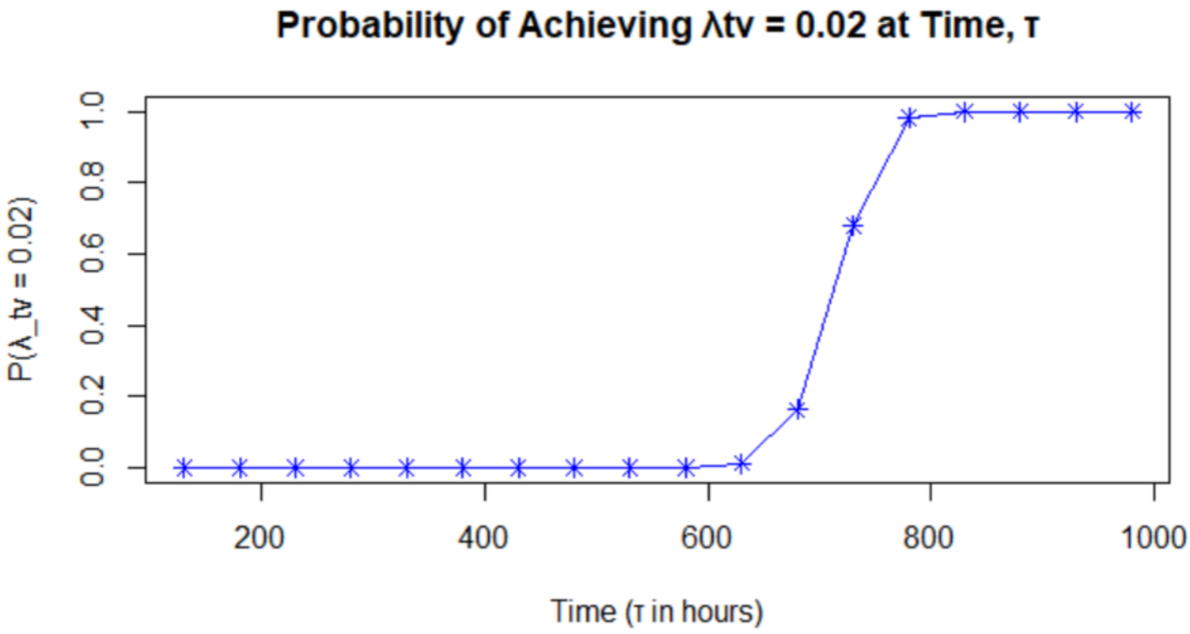

Figure 4.7: The probability of achieving $\lambda_{tv} = 0.02$ at different time periods.

Another issue of interest would be to determine the most probable failure intensity at $\tau = 500h$ since the targeted value of 0.02 ($\lambda_{tv} = 0.02$) was not achieved. An illustration was performed for the case when β is unknown to assess the relationship between failure intensity and probability while holding the time, τ, at $500h$. The study obtained the results displayed in Figure 4.8. Holding $\tau = 500h$ constant implied restricting the time window to 0-500h, such that the probabilities in the vertical axis are of achieving the failure intensities in the horizontal axis in Figure 4.8 within this time interval. For instance, a failure intensity of 0.20 has a corresponding probability of 1.0 (see Figure 4.8). This means that this failure intensity (0.20) can be achieved at any time between 0h and 500h. The Delayed S-shaped model assumes errors are corrected immediately upon detection, with no further errors introduced into the software (Hanagal & Bhalerao, 2018). As testing time increases, failure intensity would be expected to decrease steadily. It means that at $\tau = 500h$, the achievable failure intensity corresponds to those with probabilities less than 1.0 ($p < 1.0$). As such, in selecting the highest achievable failure intensity, a probability that tends to one was chosen, while the failure intensities at which the probability is exactly one (1) were disregarded. It can be observed that it is unlikely that the software will have a failure intensity of less than 0.05 at $\tau = 500h$. It can also be observed that the most probable failure intensity is between 0.06 and 0.13.

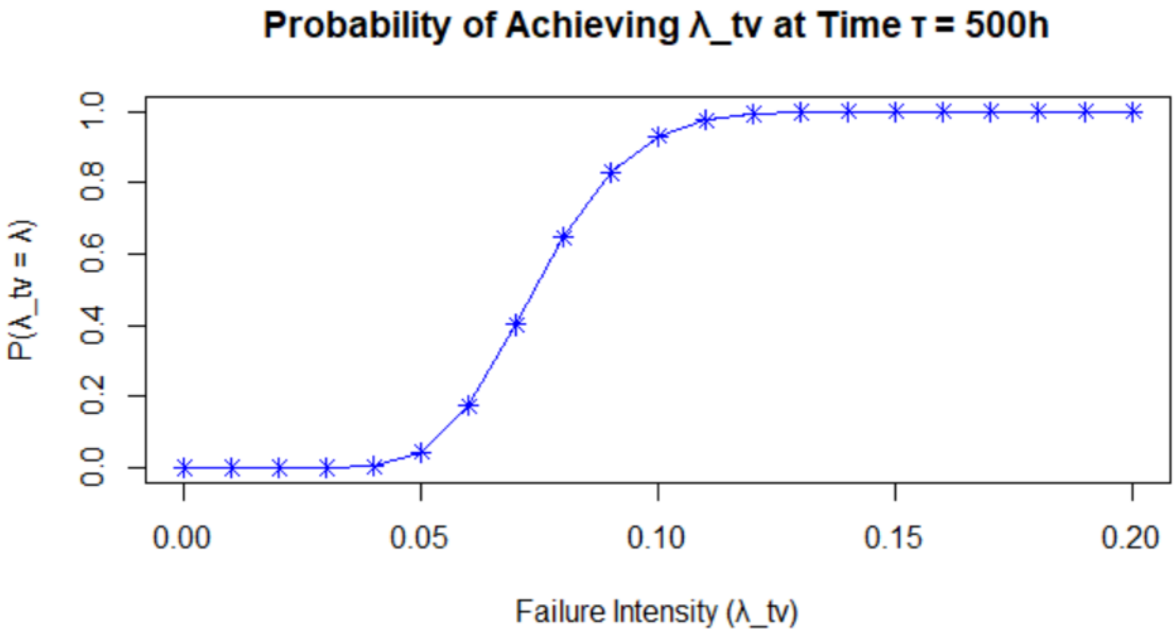

Figure 4.8: The probability of achieving $\lambda_{tv} = \lambda$ at $\tau = 500h$.

Proposition II: The next step involved determining how long it will require to achieve the target value, $\lambda_{tv} = 0.02$, since it was not attained at time $T = 100h$. (i) When β is known ($\beta = 0.007609807$), and γ is 0.90, $\tau^* = 583.365h$ was obtained using the first formula in Proposition II. The result implies that it will take another $583.365h$ to achieve the desired failure rate of 0.02 (i.e., the targeted failure intensity will be achieved at $\tau = 683.365h$). (ii) When β is unknown, $\tau^* = 657h$ was obtained using the second formula in Proposition II and Remark 1. Thus, it will require 657 additional hours to achieve the targeted failure intensity of 0.02. The results imply that when β is known, the targeted failure intensity will be achieved at 683.365h, and when it is unknown, the predicted time is 757h.

Proposition III: Since the study has established that the probable software failure intensity at a future time interval $(100, 500]h$, is high, the probability that at most k failures will occur at a future time interval $(T, \tau^*]$ was obtained, where $T < \tau^* < \tau$. Suppose the interest is to determine the probability γ_k that at most k failures will occur in the future time interval $(T, \tau^*] = (100, 130]h$, the following results were obtained for $k = 25$:

(i). When β is known ($\beta = 0.007609807$), from the first formula in Proposition III, the following results were obtained: $\gamma_0 = 0.000213$, $\gamma_1 = 0.00171$, $\gamma_2 = 0.00720$, $\gamma_3 = 0.02122$, $\gamma_4 = 0.04916$, $\gamma_5 = 0.09552$, $\gamma_6 = 0.1621$, $\gamma_7 = 0.2470$, $\gamma_8 = 0.3452$, $\gamma_9 = 0.4497$, $\gamma_{10} = 0.5529$, $\gamma_{11} = 0.6488$, $\gamma_{12} = 0.7329$, $\gamma_{13} = 0.8031$, $\gamma_{14} = 0.8590$, $\gamma_{15} =$

0.9019, $\gamma_{16} = 0.9335$, $\gamma_{17} = 0.9560$, $\gamma_{18} = 0.9716$, $\gamma_{19} = 0.9821$, $\gamma_{20} = 0.9889$, $\gamma_{21} = 0.9933$, $\gamma_{22} = 0.9960$, $\gamma_{23} = 0.9977$, $\gamma_{24} = 0.9987$, and $\gamma_{25} = 0.9992$.

(ii). When β is unknown, using the second formula in Proposition III, $\gamma_0 = 0.000024$, $\gamma_1 = 0.000230$, $\gamma_2 = 0.00113$, $\gamma_3 = 0.00391$, $\gamma_4 = 0.01054$, $\gamma_5 = 0.02375$, $\gamma_6 = 0.04651$, $\gamma_7 = 0.08137$, $\gamma_8 = 0.1298$, $\gamma_9 = 0.1916$, $\gamma_{10} = 0.2649$, $\gamma_{11} = 0.3466$, $\gamma_{12} = 0.4327$, $\gamma_{13} = 0.5189$, $\gamma_{14} = 0.6014$, $\gamma_{15} = 0.6772$, $\gamma_{16} = 0.7444$, $\gamma_{17} = 0.8018$, $\gamma_{18} = 0.8495$, $\gamma_{19} = 0.8880$, $\gamma_{20} = 0.9182$, $\gamma_{21} = 0.9413$, $\gamma_{22} = 0.9586$, $\gamma_{23} = 0.9713$, $\gamma_{24} = 0.9804$, and $\gamma_{25} = 0.9868$.

The above probabilities were plotted in Figure 9, with k (the number of software failures) on the horizontal axis and probabilities on the vertical axis. Figure 4.9 indicates that the probabilities are higher when β is known. The graphs converge at the maximum (p = 1.0), a point where it is certain that at most k software failures will be achieved within the specified time interval. From the figure, it is most likely that at most 25 failures will occur within the interval for the cases of known and unknown β. However, it is also likely that less than 25 failures will occur within the specified time interval for the case of known β.

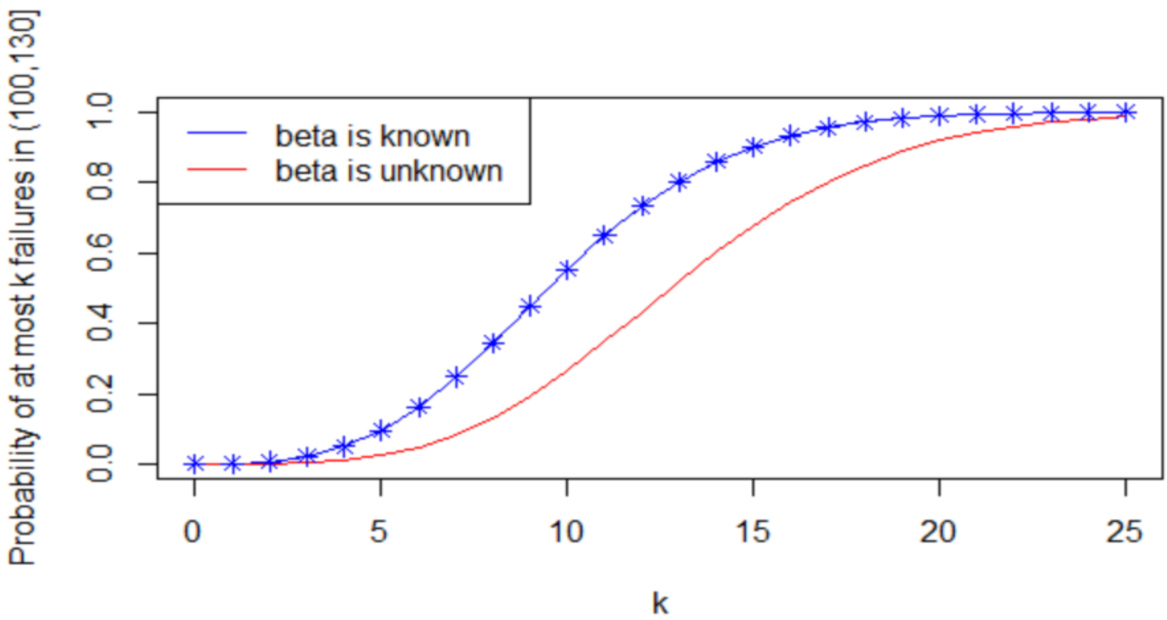

Figure 4.9: The graph of the probabilities γ_k that at most k failures will occur in the time interval $(100, 130]$ for the cases of known and unknown β.

Proposition IV: The Bayesian UPL of $\lambda(t) = \alpha\beta^2 te^{-\beta t}$ given $\tau = 700h$ and the level, $\gamma = 0.90$: (i) When β is known ($\beta = 0.0076098073$), the Bayesian UPL was obtained using the first formula in Proposition IV as $\lambda_U^{(\beta)}(\tau) = 0.01805$. There is a 95% probability that the failure intensity at $700h$ will be less than or equal to 0.01805. (ii) When β is unknown, using the second part of the formula in Proposition IV and Remark 2, the Bayesian UPL of $\lambda(t) = \alpha\beta^2 te^{-\beta t}$ with level $\gamma = 0.90$ was obtained as $\lambda_U^{(\beta)}(\tau) = 0.02883$. There is a 95% probability that the failure intensity at $700h$ will be less than or equal to 0.02883. If the target is to release the software into the market when the failure intensity is at most 0.02 ($\lambda_{tv} \leq 0.02$), it would be an optimal decision to terminate the testing process at a time when the Bayesian UPL is less than 0.02.

4.5.2 Results Obtained Using Informative Prior

The hyperparameters of the joint gamma-distributed prior were chosen arbitrarily as $a = 2$, $b = 0.005$, $c = 2$, and $d = 0.005$.

Proposition I.1: Given a predetermined failure intensity of 0.02 and time period, $\tau = 500h$, when β is known, $\gamma = 8.0 \times 10^{-9}$. Therefore, it is unlikely that the targeted failure intensity will be achieved at $\tau = 500h$. When β is unknown, $\gamma = 4.7 \times 10^{-7}$, indicating it is also unlikely that the targeted value will be achieved.

Since it is unlikely that the targeted failure intensity will be achieved at $\tau = 500h$, the study varied τ to determine the probability of attaining $\lambda_{tv} = 0.02$ at different time periods. This was illustrated for the case when β is unknown, varying time as $\tau = 130{:}\,1000, by$ 50. Figure 4.10 shows that the targeted software failure intensity would not be achieved at $630h$ or less. It can also be observed that between $630h$ and $780h$, the targeted failure intensity is achievable. At a time period $\tau \geq 780h$, it is almost certain that the targeted value will be achieved.

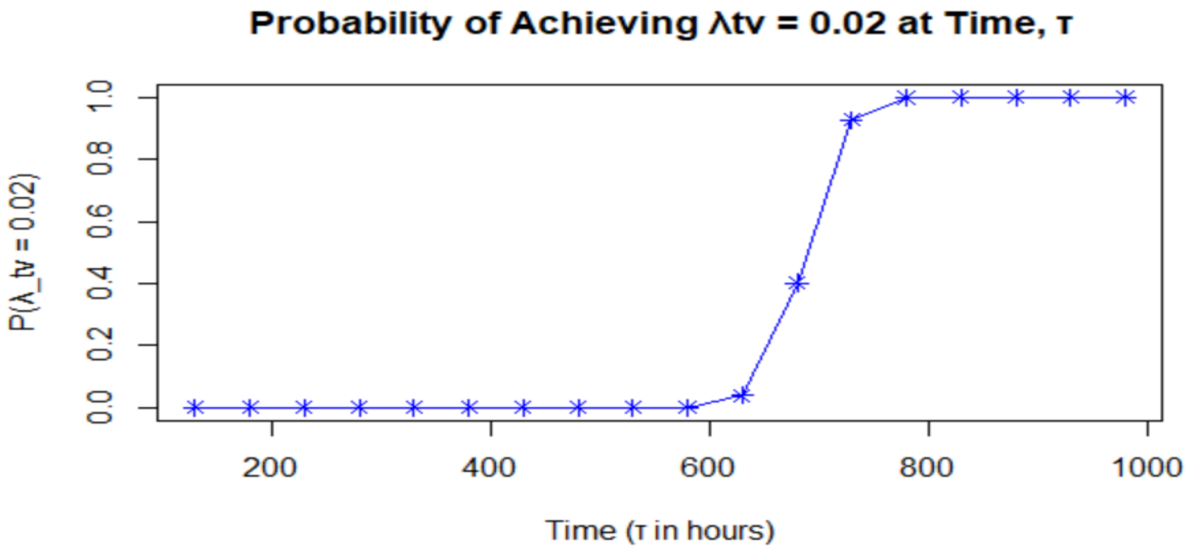

Figure 4.10: The probability of achieving $\lambda_{tv} = 0.02$ at different time periods.

Since the targeted software failure intensity ($\lambda_{tv} = 0.02$) was not achieved at $\tau = 500h$, the study determined the most likely failure intensity at this time period by varying λ_{tv} and computing their corresponding probabilities while holding $\tau = 500h$ constant. An illustration was performed for the case when β is unknown. Figure 4.11 indicates that it is unlikely to achieve a software failure intensity of less than 0.05 at $\tau = 500h$. It can be observed that the achievable failure intensity lies between $\lambda = 0.05$ and $\lambda = 0.10$.

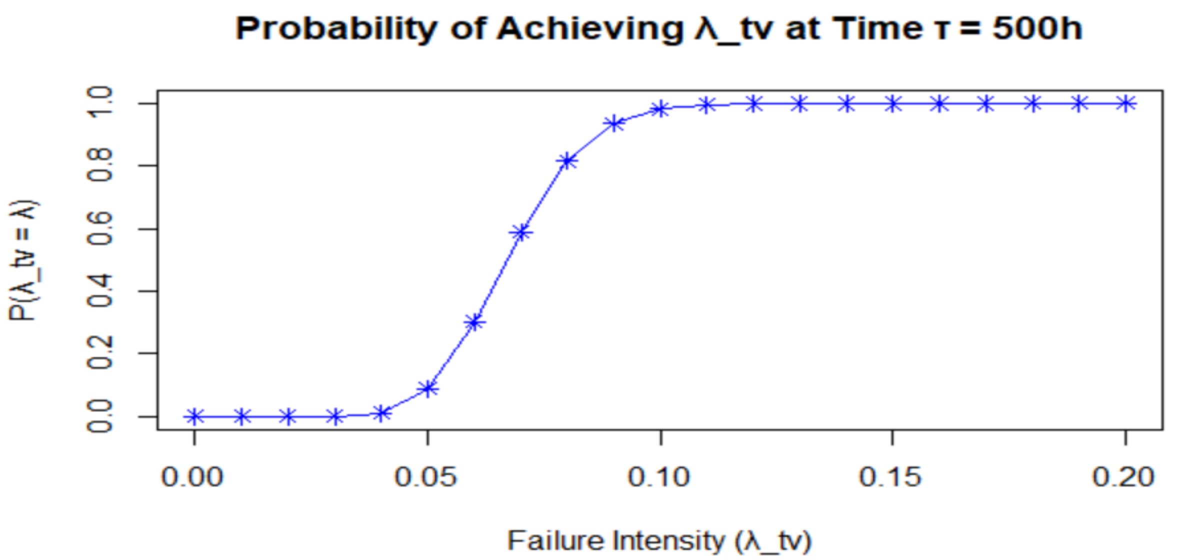

Figure 4.11: The probability of achieving $\lambda_{tv} = \lambda$ at $\tau = 500h$.

Proposition II.1: Since the targeted failure intensity was not achieved at $\tau = 500h$, the study determined how long it would take to achieve this failure intensity. (i) When β is known and $\gamma = 0.90$, using the first formula in Proposition II.1, the additional time, $\tau^* = 578.8357h$. The result implies that it will take another $578.8357h$ to attain the target value, $\lambda_{tv} = 0.02$: The failure intensity is predicted to be achieved at time $678.835h$. (ii) When β is unknown, $\tau^* = 627h$, implying it will take additional $627h$ to achieve the targeted failure intensity. That is, it would be achieved at $\tau = 727h$. These results may inform software developers about the predicted time at which they expect to achieve the targeted failure intensity, hence the decision to terminate the testing process and timely release of the software into the market.

Proposition III.1: Suppose the interest is to obtain the probabilities γ_k that at most k failures will occur in the future time period $(100, 130]h$. The following results were obtained for $k = 25$, and plotted in Figure 4.12.

(i). When β is known ($\beta = 0.007609807$), using the first formula in Proposition III.1, we have $\gamma_0 = 0.000122, \gamma_1 = 0.00104, \gamma_2 = 0.00463, \gamma_3 = 0.01437, \gamma_4 = 0.03494, \gamma_5 = 0.07101$, $\gamma_6 = 0.1256$, $\gamma_7 = 0.1988$, $\gamma_8 = 0.2875$, $\gamma_9 = 0.3863$, $\gamma_{10} = 0.4884$, $\gamma_{11} = 0.5871$, $\gamma_{12} = 0.6773$, $\gamma_{13} = 0.7554$, $\gamma_{14} = 0.8200$, $\gamma_{15} = 0.8713$, $\gamma_{16} = 0.9104$, $\gamma_{17} = 0.9392$, $\gamma_{18} = 0.9597$, $\gamma_{19} = 0.9739$, $\gamma_{20} = 0.9835$, $\gamma_{21} = 0.9897$, $\gamma_{22} = 0.9938$, $\gamma_{23} = 0.9963$, $\gamma_{24} = 0.9978$, and $\gamma_{25} = 0.9987$.

(ii). When β is unknown, using the second formula in Proposition III.1, we get $\gamma_0 = 0.0000634$, $\gamma_1 = 0.000568$, $\gamma_2 = 0.00266$, $\gamma_3 = 0.00866$, $\gamma_4 = 0.0221$, $\gamma_5 = 0.04705$, $\gamma_6 = 0.08702$, $\gamma_7 = 0.1438$, $\gamma_8 = 0.2168$, $\gamma_9 = 0.3028$, $\gamma_{10} = 0.3969$, $\gamma_{11} = 0.4934$, $\gamma_{12} = 0.5866$, $\gamma_{13} = 0.6722$, $\gamma_{14} = 0.7472$, $\gamma_{15} = 0.8102$, $\gamma_{16} = 0.8611$, $\gamma_{17} = 0.9009$, $\gamma_{18} = 0.9308$, $\gamma_{19} = 0.9528$, $\gamma_{20} = 0.9685$, $\gamma_{21} = 0.9794$, $\gamma_{22} = 0.9867$, $\gamma_{23} = 0.9916$, $\gamma_{24} = 0.9948$, and $\gamma_{25} = 0.9968$.

Figure 4.12 shows that the probabilities are slightly higher when β is known. The two graphs converge slightly below the maximum (p = 1.0) when approaching the 25$^{\text{th}}$ failure. The pattern suggests that it is most likely that at most 25 failures will occur within the specified time interval. This information is appropriate for software developers in planning tests as they will prepare for at most 25 failures in the time interval $(100, 130]h$.

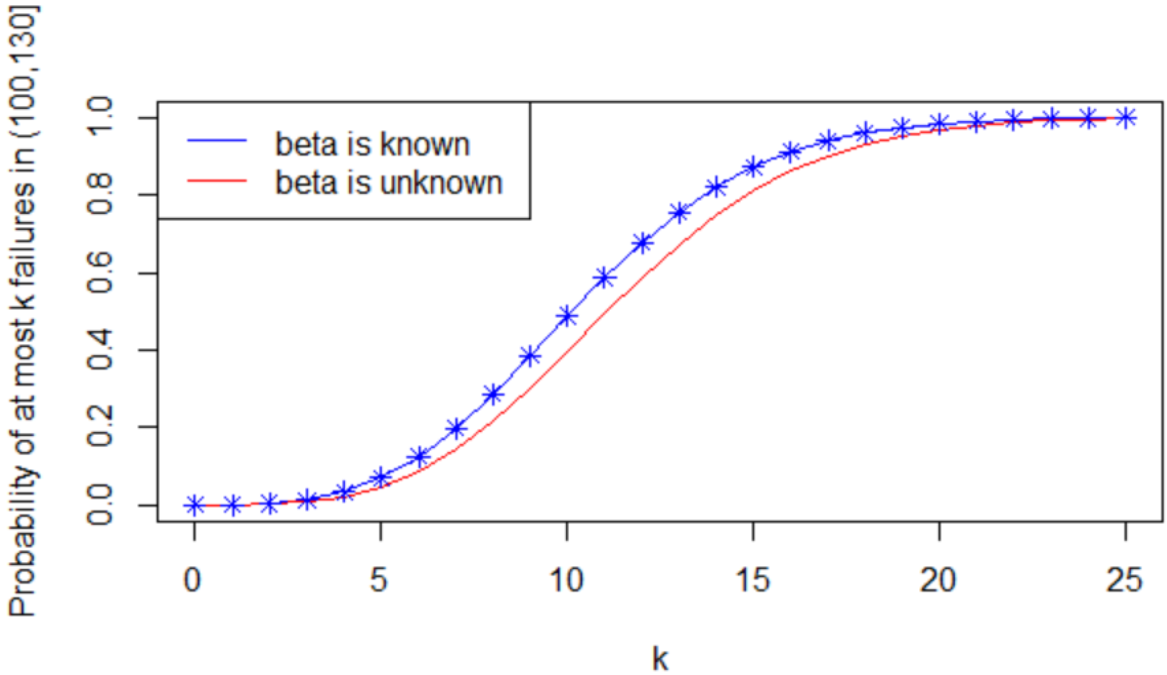

Figure 4.12: The graph of the probabilities γ_k that at most k failures will occur in the time interval $(100, 130]$ for the cases of known and unknown β.

Proposition IV.1: Suppose the interest is to obtain the Bayesian UPL of $\lambda(t) = \alpha\beta^2 te^{-\beta t}$ given $\tau = 700h$ and the level $\gamma = 0.90$; (i) When β is known ($\beta = 0.007609807$), the Bayesian UPL was obtained using the first formula in Proposition IV.1 as $\lambda_U^{(\beta)}(\tau) = 0.01756$. (ii) When β is unknown, the Bayesian UPL of $\lambda(t)$ was obtained as $\lambda_U^{(\beta)}(\tau) = 0.02392$. The failure intensity at $\tau = 700h$ is predicted to be less than or equal to 0.01756 when β is known and less than or equal to 0.02392 when β is unknown, with a 95% probability.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

This chapter provides conclusions from the analyses performed. Recommendations based on the results and for further research are also provided.

5.1 Conclusions

In most software reliability assessment problems, researchers aim to obtain accurate parameter estimates using small data available to enhance accuracy in prediction and save on resources. The primary objective of this study was to perform Bayesian interval estimation, launch a comparison with Wald confidence intervals, and address prediction issues in software reliability testing. Interval lengths and coverage probabilities were used for comparison. The following conclusions have been drawn.

i. The study successfully constructed Bayesian credible intervals using gamma-distributed informative prior and $1/\alpha\beta$ and $1/\alpha$ non-informative priors.
ii. The Bayesian methods with gamma-distributed informative and $1/\alpha\beta$ priors yielded more precise interval estimates than the Wald approach. However, the method with $1/\alpha$ joint prior yielded unreasonably wider intervals, performing poorly compared to Wald interval estimates.
iii. The study considered the three priors and addressed four prediction issues, including determining the probability of achieving a specified failure intensity at a time $\tau, \tau > T$, determining the time τ^* required to achieve the specified failure intensity if it was not attained at $T = 100h$, determining the probability of at most k failures in the future time period (T, τ), and obtaining the Bayesian UPL. The study developed methods for addressing these software reliability problems and successfully illustrated them using secondary software failure data.

5.2 Recommendations

This thesis has considered Bayesian interval estimation and predictive analysis on the Delayed S-shaped software reliability model. The developed methods can be used in software quality assessment to predict software reliability and the optimal time to terminate the testing process. Future researchers can extend the developed procedures to other NHPP models, such as the Cox-Lewis process, the Yamada imperfect debugging model, and the extended Delayed S-shaped model that captures imperfect debugging. Moreover, the study has derived posterior and predictive distributions for one-sample software on the Delayed S-shaped model to address four prediction issues that may be of interest during software testing. However, two-sample prediction on the model may be explored using the procedures in the study by Yu *et al.* (2007).

REFERENCES

Achcar, J. A., Dey, D. K., & Niverthi, M. (1998). A Bayesian approach using nonhomogeneous Poisson processes for software reliability models. *Series on Quality, Reliability and Engineering Statistics Frontiers in Reliability*, 4, 1-18. https://doi.org/10.1142/9789812816580_0001

Akuno, A. O., Orawo, L. A. O., & Islam, A. S. (2014a). Two-sample Bayesian predictive analyses for an exponential non-homogeneous Poisson process in software reliability. *Open Journal of Statistics*, *4*(09), 742. http://dx.doi.org/10.4236/ojs.2014.49069

Akuno, A. O., Orawo, L. A., & Islam, A. S. (2014b). One-sample Bayesian predictive analyses for an exponential nonhomogeneous Poisson process in software reliability. *Open Journal of Statistics*, *4*(5), 402-411. http://dx.doi.org/10.4236/ojs.2014.45039

Anniprincy, B., & Sridhar, S. (2014). Two dimensional software reliability growth models using Cobb-Douglas production function and Yamada S-shaped model. *Journal of Software Engineering Simulation*, *2*(2), 1-11.

Azzolini, D., Riguzzi, F., Masotti, F., & Lamma, E. (2019). A comparison of MCMC sampling for probabilistic logic programming. In *International Conference of the Italian Association for Artificial Intelligence*. *Springer International Publishing*, 18-29. https://doi.org/10.1007/978-3-030-35166-3_2

Bai, C. (2005). Bayesian network-based software reliability prediction with an operational profile. *Journal of Systems and Software*, *77*(2), 103-112. https://doi.org/10.1016/j.jss.2004.11.034

Berger, J. O., Bernardo, J. M., & Sun, D. (2009). The formal definition of reference priors. *The Annals of Statistics*, *37*(2), 905-938. https://doi.org/10.1214/07-AOS587

Cheruiyot, N., Orawo, L. A. O., & Islam, A. S. (2018). Bayesian predictive analyses for logarithmic non-homogeneous Poisson process in software reliability. *Open Access Library Journal*, *5*(8), 1-13. https://doi.org/10.4236/oalib.1104767

Cheruiyot, N., Orawo, L. A. O., & Islam, A. S. (2019). Predictive analyses of logarithmic non-homogeneous Poisson process in software reliability using Bayesian approach with informative priors. *American Journal of Mathematics and Statistics*, *9* (2), 57-65. http://article.sapub.org/10.5923.j.ajms.20190902.02.html

Chulani, S., Boehm, B., & Steece, B. (1999). Bayesian analysis of empirical software engineering cost models. *IEEE Transactions on Software Engineering*, *25*(4), 573-583. https://doi.org/10.1109/32.799958

Cid, J. E. R., & Achcar, J. A. (1999). Bayesian inference for nonhomogeneous Poisson processes in software reliability models assuming nonmonotonic intensity functions. *Computational Statistics & Data Analysis*, *32*(2), 147-159. https://doi.org/10.1016/S0167-9473(99)00028-6

Cox, D. R., & Lewis, P. A. (1966). *The statistical analysis of series of events*. https://link.springer.com/book/9789401178037

Dalitz, C. (2018). Construction of Confidence Intervals. *Niederrhein University of Applied Sciences, 1,* 15-28. https://doi.org/10.48550/arXiv.1807.03582

Dodwell, T. J., Ketelsen, C., Scheichl, R., & Teckentrup, A. L. (2015). A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *SIAM/ASA Journal on Uncertainty Quantification*, *3*(1), 1075-1108. https://doi.org/10.1137/130915005

Duane, J. T. (1964). Learning curve approach to reliability monitoring. *IEEE transactions on Aerospace*, *2*(2), 563-566. https://doi.org/10.1109/TA.1964.4319640

Ehrlich, W., Prasanna, B., Stampfel, J., & Wu, J. (1993). Determining the cost of a stop-test decision (software reliability). *IEEE Software*, *10*(2), 33-42. https://doi.org/10.1109/52.199726

Glickman, M. E., & Van Dyk, D. A. (2007). Basic Bayesian methods. *Topics in Biostatistics*, *404*, 319-338. https://doi.org/10.1007/978-1-59745-530-5_16

Goel, A. L., & Okumoto, K. (1979). Time-dependent error-detection rate model for software reliability and other performance measures. *IEEE Transactions on Reliability*, *28*(3), 206-211. https://doi.org/10.1109/TR.1979.5220566

Golchi, S. (2018). Informative priors in Bayesian inference and computation. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *12*(2), 45–55. https://doi.org/10.1002/sam.11371

Gupta, A., Choudhary, D., & Saxena, S. (2011). Software reliability estimation using Yamada delayed S-shaped model under imperfect debugging and time lag. *International Journal of Computer Applications*, *23*(7), 49-54. https://doi.org/10.5120/2985-3792

Hanagal, D. D., & Bhalerao, N. N. (2018). Analysis of delayed S-shaped software reliability growth model with time dependent fault content rate function. *Journal of Data Science*, *16*(4), 857-878. https://doi.org/10.6339/jds.201810_16(4).00010

Hirata, T., Okamura, H., & Dohi, T. (2009). A Bayesian inference tool for NHPP-based software reliability assessment. In *Future Generation Information Technology: First International Conference, FGIT 2009, Jeju Island, Korea, December 10-12, 2009. Proceedings 1*, 225-236.

Hung-Cuong, N., & Quyet-Thang, H. (2015). New NHPP SRM based on generalized S-shaped fault-detection rate function. In *Nature of Computation and Communication: International Conference, ICTCC 2014, Ho Chi Minh City, Vietnam, November 24-25, 2014, Revised Selected Papers 1* (pp. 212-221). Springer International Publishing. https://doi.org/10.1007/978-3-319-15392-6_21

Inoue, S., & Yamada, S. (2006). Discrete software reliability assessment with discretized NHPP models. *Computers & Mathematics with Applications*, *51*(2), 161-170. https://doi.org/10.1016/j.camwa.2005.11.022

Ismael Al turk, L. (2019). Confidence interval estimation of the conditional reliability function for time domain data. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, *52*(1), 197-208.

Jelinski, Z., & Moranda, P. (1972). Software reliability research. In *Statistical Computer Performance Evaluation*, 465-484. https://doi.org/10.1016/B978-0-12-266950-7.50028-1

Johannes, M., & Polson, N. (2010). MCMC methods for continuous-time financial econometrics. In *Handbook of Financial Econometrics: Applications* (pp. 1-72). Elsevier. https://doi.org/10.1016/B978-0-444-53548-1.50003-9

Kaur, R., & Panwar, P. (2015). Study of perfect and imperfect debugging NHPP SRGMs used for prediction of faults in a software. *International Journal of Computer Science & Communications*, *6*(1), 73-78.

Kimura, M., Yamada, S., & Osaki, S. (1992). Software reliability assessment for an exponential-S-shaped reliability growth phenomenon. *Computers & Mathematics with Applications*, *24*(1-2), 71-78. https://doi.org/10.1016/0898-1221(92)90230-F

Kruschke, J. K. (2010). What to believe: Bayesian methods for data analysis. *Trends In Cognitive Sciences*, *14*(7), 293-300. https://doi.org/10.1016/j.tics.2010.05.001

Kuo, L., & Yang, T. Y. (1996). Bayesian computation for nonhomogeneous Poisson processes in software reliability. *Journal of the American Statistical Association, 91*(434), 763-773. https://doi.org/10.1080/01621459.1996.10476944

Kuo, L., Lee, J. C., Choi, K., & Yang, T. Y. (1997). Bayes inference for S-shaped software-reliability growth models. *IEEE Transactions on Reliability, 46*(1), 76-80. https://doi.org/10.1109/24.589931

Lai, R., & Garg, M. (2012). A detailed study of NHPP Software Reliability Models (invited paper). *Journal of Software*, *7*(6). https://doi.org/10.4304/jsw.7.6.1296-1306

Lee, M. D., & Vanpaemel, W. (2018). Determining informative priors for cognitive models. *Psychonomic Bulletin & Review*, *25* (1), 114-127. https://doi.org/10.3758/s13423-017-1238-3

Lee, T. Q., Yeh, C. W., & Fang, C. C. (2014). Bayesian software reliability prediction based on Yamada delayed S-shaped model. *Applied Mechanics and Materials*, *490–491*, 1267–1278. https://doi.org/10.4028/www.scientific.net/amm.490-491.1267

Lewis, P. W., & Shedler, G. S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly*, *26*(3), 403-413. https://doi.org/10.1002/nav.3800260304

Martino, L., Elvira, V., & Camps-Valls, G. (2018). The recycling Gibbs sampler for efficient learning. *Digital Signal Processing*, *74*, 1-13. https://doi.org/10.1016/j.dsp.2017.11.012

Musa, J. D., & Okumoto, K. (1984). A logarithmic Poisson execution time model for software reliability measurement. In *Proceedings of the 7th international conference on Software engineering*, 230-238. https://doi.org/10.5555/800054.801975

Petzschner, F. H., Glasauer, S., & Stephan, K. E. (2015). A Bayesian perspective on magnitude estimation. *Trends in Cognitive Sciences*, *19* (5), 285-293. https://doi.org/10.1016/j.tics.2015.03.002

Pham, H., & Zhang, X. (2003). NHPP software reliability and cost models with testing coverage. *European Journal of Operational Research*, *145* (2), 443-454. https://doi.org/10.1016/s0377-2217(02)00181-9

Pham, L., & Pham, H. (2000). Software reliability models with time-dependent hazard function based on Bayesian approach. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *30*(1), 25-35. https://doi.org/10.1109/3468.823478

Rahman, A., Gao, J., D'Este, C., & Ahmed, S. E. (2016). An assessment of the effects of prior distributions on the Bayesian predictive inference. *International Journal of Statistics and Probability*, *5*(5), 31-42. https://doi.org/10.5539/ijsp.v5n5p31

Robert, C. P., Casella, G., Robert, C. P., & Casella, G. (2004). The Metropolis—Hastings algorithm. *Monte Carlo Statistical Methods*, 267-320. https://doi.org/10.1007/978-1-4757-4145-2_7

Robles, N. T. (2011). *Stochastic comparisons and Bayesian inference in software reliability* [Doctoral dissertation, Universidad Carlos III de Madrid].

Song, K. Y., Chang, I. H., & Pham, H. (2017). An NHPP software reliability model with S-shaped growth curve subject to random operating environments and optimal release time. *Applied Sciences*, *7*(12), 1304. https://doi.org/10.3390/app7121304

Tian, G. L., Tang, M. L., & Yu, J. W. (2011). Bayesian estimation and prediction for the power law process with left-truncated data. *Journal of Data Science, 9*(3), 445-470. https://doi.org/10.6339/jds.201107_09(3).0009

van de Schoot, R., Depaoli, S., King, R., Kramer, B., Märtens, K., Tadesse, M. G., Vannucci, M., Willemsen, J., & Yau, C. (2021). Bayesian statistics and modelling. *Nature Reviews Methods Primers*, *1* (1), 1-26. https://doi.org/10.1038/s43586-020-00003-0

Wahono, R. S. (2015). A systematic literature review of software defect prediction. *Journal of Software Engineering*, *1*(1), 1-16.

Xie, M., Hu, Q. P., Wu, Y. P., & Ng, S. H. (2007). A study of the modeling and analysis of software fault-detection and fault-correction processes. *Quality and Reliability Engineering International*, *23*(4), 459-470. https://doi.org/10.1002/qre.827

Yamada, S., Ohba, M., & Osaki, S. (1983). S-shaped reliability growth modeling for software error detection. *IEEE Transactions on Reliability*, *32*(5), 475-484. https://doi.org/10.1109/TR.1983.5221735

Yamada, S., Ohba, M., & Osaki, S. (1984). S-shaped software reliability growth models and their applications. *IEEE Transactions on Reliability, 33*(4), 289-292. https://doi.org/10.1109/TR.1984.5221826

Yamada, S., Osaki, S., & Narihisa, H. (1985). A software reliability growth model with two types of errors. *RAIRO-Operations Research*, *19*(1), 87-104. https://doi.org/10.1051/ro/1985190100871

Yin, L., & Trivedi, K. S. (1999). Confidence interval estimation of NHPP-based software reliability models. In *Proceedings 10th International Symposium on Software*

Reliability Engineering (Cat. No. PR00443), IEEE, 6-11. https://doi.org/10.1109/ISSRE.1999.809305

Yoo, T. H. (2018). A performance valuation for NHPP software reliability model depend on Weibull-type distribution. *International Journal of Pure and Applied Mathematics, 118*(19), 1021-1032.

Yu, J. W., Tian, G. L., & Tang, M. L. (2007). Predictive analyses for nonhomogeneous Poisson processes with power law using Bayesian approach. *Computational Statistics & Data Analysis*, *51*(9), 4254-4268. https://doi.org/10.1016/j.csda.2006.05.010

Zhao, Z., Zhang, Y., Liu, G., & Qiu, J. (2017). Statistical analysis of time-varying characteristics of testability index based on NHPP. *IEEE Access*, *5*, 4759-4768. https://doi.org/10.1109/ACCESS.2017.2688698

APPENDICES

Appendix A: Publications

Appendix A1: Research Publication 1

American Journal of Theoretical and Applied Statistics
2023; 12(3): 43-50
http://www.sciencepublishinggroup.com/j/ajtas
doi: 10.11648/j.ajtas.20231203.12
ISSN: 2326-8999 (Print); ISSN: 2326-9006 (Online)

SciencePG
Science Publishing Group

Bayesian Interval Estimation in a Non-Homogeneous Poisson Process with Delayed S-Shaped Intensity Function

Otieno Collins, Orawo Luke Akong'o, Matiri George Munene, Justin Obwoge Okenye

Department of Mathematics, Egerton University, Egerton, Kenya

Email address:

Otienoc292@gmail.com (Otieno Collins), luke.orawo@egerton.ac.ke (Orawo Luke Akong'o), gemuwa@gmail.com (Matiri George Munene) justinoke69@gmail.com (Justin Obwoge Okenye)

To cite this article:

Otieno Collins, Orawo Luke Akong'o, Matiri George Munene, Justin Obwoge Okenye, Bayesian Interval Estimation in a Non-Homogeneous Poisson Process with Delayed S-Shaped Intensity Function. *American Journal of Theoretical and Applied Statistics*. Vol. 12, No. 3, 2023, pp. 43-50. doi: 10.11648/j.ajtas.20231203.12

Received: April 11, 2023; **Accepted**: May 16, 2023; **Published**: June 5, 2023

Abstract: Software reliability assessment has been explored by many researchers over the past decades. With the increasing development of new complex software systems, accurate methods for estimating reliability model parameters are needed. Facilitated by the increasing use of computer systems in various sectors such as air traffic control, banking, industrial processes, and government operations, developing accurate reliability assessment methods is indispensable. The Delayed S-shaped software reliability model is one of the non-homogeneous Poisson process (NHPP) software reliability models proposed for capturing error detection and removal processes in software reliability testing. Many researchers have fitted the model to software failure data and performed estimation using the Maximum Likelihood method and Bayesian approach, however, construction of Bayesian credible sets for the parameters of this model and comparison of their efficiencies with the Wald confidence intervals using simulation have not been explored. The Bayesian interval estimation was conducted with three different joint prior distributions assigned to the parameters α and β of the model, namely the gamma distributed informative prior and, $1/\alpha$, and $1/\alpha\beta$ as non-informative priors. The Bayesian credible intervals and Wald confidence intervals for the two parameters were compared on the basis of interval lengths and coverage probabilities. The simulation was assumed to emulate the end-user environment and can generate inter-failure times data for the study. The Delayed S-shaped reliability model variables were simulated with fixed parameters set at (α, β)=(20, 0.5). The hyperparameters for the informative prior were chosen such that they have minimal effect on the results. In other words, the prior information does not swamp the information from the data. The Bayesian method yields superior results, as evidenced by shorter interval lengths and higher coverage probabilities in Table 1.

Keywords: Non-Homogeneous Poisson Process, Intensity Function, Software Reliability Model, Informative Priors, Bayesian Method, Wald Intervals, Maximum Likelihood

Appendix A2: Research Publication 2

Scientific Research Publishing

Open Journal of Statistics, 2023, 13, 717-733
https://www.scirp.org/journal/ojs
ISSN Online: 2161-7198
ISSN Print: 2161-718X

One-Sample Bayesian Predictive Analyses for a Nonhomogeneous Poisson Process with Delayed S-Shaped Intensity Function Using Non-Informative Priors

Otieno Collins, Orawo Luke Akong'o, Matiri George Munene

Department of Mathematics, Egerton University, Egerton, Kenya
Email: otienoc292@gmail.com, luke.orawo@egerton.ac.ke, gemawa@gmail.com

How to cite this paper: Collins, O., Akong'o, O.L. and Munene, M.G. (2023) One-Sample Bayesian Predictive Analyses for a Nonhomogeneous Poisson Process with Delayed S-Shaped Intensity Function Using Non-Informative Priors. *Open Journal of Statistics*, **13**, 717-733.
https://doi.org/10.4236/ojs.2023.135034

Received: July 19, 2023
Accepted: October 7, 2023
Published: October 10, 2023

Abstract

The delayed S-shaped software reliability growth model (SRGM) is one of the non-homogeneous Poisson process (NHPP) models which have been proposed for software reliability assessment. The model is distinctive because it has a mean value function that reflects the delay in failure reporting: there is a delay between failure detection and reporting time. The model captures error detection, isolation, and removal processes, thus is appropriate for software reliability analysis. Predictive analysis in software testing is useful in modifying, debugging, and determining when to terminate software development testing processes. However, Bayesian predictive analyses on the delayed S-shaped model have not been extensively explored. This paper uses the delayed S-shaped SRGM to address four issues in one-sample prediction associated with the software development testing process. Bayesian approach based on non-informative priors was used to derive explicit solutions for the four issues, and the developed methodologies were illustrated using real data.

Keywords

Failure Intensity, Non-Informative Priors, Software Reliability Model, Bayesian Approach, Non-Homogeneous Poisson Process

Appendix B: Research Permit

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **904999**

Date of Issue: **11/May/2023**

RESEARCH LICENSE

This is to Certify that Mr.. Collins Otieno Omenda of Egerton University, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Nakuru on the topic: BAYESIAN INTERVAL ESTIMATION AND PREDICTIVE ANALYSIS IN A NONHOMOGENEOUS POISSON PROCESS WITH DELAYED S-SHAPED INTENSITY FUNCTION for the period ending : 11/May/2024.

License No: **NACOSTI/P/23/25575**

904999

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions

Appendix C: Key R-codes

The following R-codes were used in data simulation and computation of the results.

Wald confidence and Bayesian credible intervals

i. Simulation of inter-failure times data

```
T0=100; Alpha=20; Beta=0.05; lambda0=0.4
Lambda=function(z)   {Alpha*(Beta^2)*z*exp(-Beta*z)}
Simm=function(Alpha,Beta,lambda0,T0){
  t=0; I=0;
  S=c()
  while(t<T0){
    U=runif(1)
    t=t-log(U)/lambda0
    V2=runif(1)
    if(V2<=Lambda(t)/lambda0){
      I=I+1
      S[I]=t}else{I=I}
  }
  return(S)
}
```

ii. Wald confidence interval

```
LoglikN2=function(par){
  alpha=par[1]
  beta=par[2]
  #data=S
  Val1=-((n*log(alpha))+(2*n*log(beta))+sum(log(S))-(beta*(sum(S)))-alpha*(1-
(1+beta*T0)*exp(-beta*T0)))
  #return(Val1)
}
parN=c(18,0.03)
mleN1=nlm(LoglikN2,parN,hessian = TRUE)
mleN1

HTM=solve(mleN1$hessian)
HTM
```

```
NWald=function(Alphahat,Betahat){
  SigAlp=sqrt(HTM[1,1])
  SigBet=sqrt(HTM[2,2])
  LowerAlp=Alphahat-(qnorm(.975))*SigAlp
  UpperAlp=Alphahat+(qnorm(.975))*SigAlp
  LowerBet=Betahat-(qnorm(.975))*SigBet
  UpperBet=Betahat+(qnorm(.975))*SigBet
  CIAlp=c(LowerAlp,UpperAlp)
  CIBet=c(LowerBet,UpperBet)
  return(list(CIAlp=CIAlp,CIBet=CIBet))
}
CIAB=NWald(mleN1$estimate[1],mleN1$estimate[2])
CIAB
Intla = diff(CIAB$CIAlp)
Intlb = diff(CIAB$CIBet).
```

iii. Bayesian credible interval using informative prior

```
BEUT=function(beta,data,alpha,c,d,T0){
  data=S
  n=length(S)
  V=2*n+c-1
  R=d+sum(S)
  KK=(1+beta*T0)
  NDC= exp(alpha*(KK*exp(-beta*T0)))
  P1=(beta^V)*NDC
  P2=exp(-beta*R)
  M=P1*P2
  return(M)
}

###MH
MetroH=function(stat,data,k,m,c,d,theta,L){
  #stat=c(18,0.03)
  data=S
  n=length(S)
  ##Storage
  beta=rep(NA,L)
  alpha=rep(NA,L)
  coun=0
  ###Initialize
  alpha[1]=stat[1]
  beta[1]=stat[2]
```

```
  for(i in 2:L){
    A=n+k
    B=(m+1-(1+beta[i-1]*T0)*exp(-beta[i-1]*T0))
    alpha[i]=rgamma(1,A,B)

    BetaProp=rnorm(1,beta[i-1],theta)
    R=(BeUT(BetaProp,data,alpha[i],c,d,T0))/(BeUT(beta[i-
1],data,alpha[i],c,d,T0))
    ##CF=(dnorm(Beta[i-1],BetaProp,theta))/(dnorm(BetaProp,Beta[i-1],theta))
    MR=R#*CF
    u=runif(1)
    if (u<MR & !is.na(MR)){beta[i]=BetaProp; coun=coun+1} else {beta[i]=beta[i-
1]}
  }
  Acc=coun/L*100
  return(list(alpha=alpha,beta=beta,Acc=Acc))
}
```

iv. The following codes were used for repeated sampling:

```
G=5000
i=0;j=0
ns=c()
intLa=c()
intLb=c()
intlA=c()
intlB=c()
ahat=c()
BB=matrix(rep(NA,11*G),ncol = 11)
DD=matrix(rep(NA,11*G),ncol = 11)
CC=matrix(rep(NA,11*G),ncol = 11)


for(k in 1:G){
  t=0; I=0;
  S=c()
  while(t<T1){
    U=runif(1)
    t=t-log(U)/lambda0
    V2=runif(1)
    if(V2<=Lambda(t)/lambda0){
```

```
    I=I+1
    S[I]=t}else{I=I}
  }
  S
  n=length(S)
  Loglik1=function(par){
    alpha=par[1]
    beta=par[2]
    Val1=-n*log(alpha)-(2*n*log(beta))-sum(log(S))+(beta*(sum(S)))+alpha*(1-
(1+beta*T1)*exp(-beta*T1))
    return(Val1)
  }
  par0=c(18,0.03)
  mle=nlm(Loglik1,par0,hessian=TRUE)
  mle

  ns[k]=length(S)
  parest=mle$estimate
  ahat=parest[1]
  A=solve(mle$hessian)
  aint=c(parest[1]-1.96*sqrt(A[1,1]),parest[1]+1.96*sqrt(A[1,1]))
  bint=c(parest[2]-1.96*sqrt(A[2,2]),parest[2]+1.96*sqrt(A[2,2]))
  intLa[k]=diff(aint)
  intLb[k]=diff(bint)
  if(aint[1]<Alpha && aint[2]>Alpha){i=i+1} else{i=i}
  if(bint[1]<Beta && bint[2]>Beta){j=j+1} else{j=j}
  n=length(S)
  BB[k,1]=n
  BM=MetroH(stat=c(18,0.03),S,2,0.005,2,0.005,0.05,5000)
  estB=c(mean(BM$alpha),mean(BM$beta))
  Alphat2=estB[1]
  Bethat2=estB[2]
  CIAlp2=quantile(BM$alpha,c(.025,.975))
  CIBet2=quantile(BM$beta,c(.025,.975))
```

```
BB[k,2]=Alphat2
BB[k,3:4]=CIAlp2
BB[k,5]=diff(CIAlp2)
if(CIAlp2[1]<Alpha && CIAlp2[2]>Alpha){BB[k,6]=1}
else{BB[k,6]=0}
BB[k,7]=Bethat2
BB[k,8:9]=CIBet2
BB[k,10]=diff(CIBet2)
if(CIBet2[1]<Beta && CIBet2[2]>Beta){BB[k,11]=1}
else{BB[k,11]=0}

DD[k,1]=n
BMN=MetroH2(stat=c(18,0.03),S,0.05,5000)
estNB=c(mean(BMN$alpha),mean(BMN$beta))
AlphatN=estNB[1]
BethatN=estNB[2]
CIAlpN=quantile(BMN$alpha,c(.025,.975))
CIBetN=quantile(BMN$beta,c(.025,.975))
DD[k,2]=AlphatN
DD[k,3:4]=CIAlpN
DD[k,5]=diff(CIAlpN)
if(CIAlpN[1]<Alpha && CIAlpN[2]>Alpha){DD[k,6]=1}
else{DD[k,6]=0}
DD[k,7]=BethatN
DD[k,8:9]=CIBetN
DD[k,10]=diff(CIBetN)
if(CIBetN[1]<Beta && CIBetN[2]>Beta){DD[k,11]=1}
else{DD[k,11]=0}
CC[k,1]=n
BMN1=MetroH3(stat=c(18,0.03),S,0.05,5000)
estNB1=c(mean(BMN1$alpha),mean(BMN1$beta))
AlphatN1=estNB1[1]
BethatN1=estNB1[2]
CIAlpN1=quantile(BMN1$alpha,c(.025,.975))
```

```
  CIBetN1=quantile(BMN1$beta,c(.025,.975))
  CC[k,2]=AlphatN1
  CC[k,3:4]=CIAlpN1
  CC[k,5]=diff(CIAlpN1)
  if(CIAlpN1[1]<Alpha && CIAlpN1[2]>Alpha){CC[k,6]=1}
  else{CC[k,6]=0}
  CC[k,7]=BethatN1
  CC[k,8:9]=CIBetN1
  CC[k,10]=diff(CIBetN1)
  if(CIBetN1[1]<Beta && CIBetN1[2]>Beta){CC[k,11]=1}
  else{CC[k,11]=0}
}
#Wald coverage probability
c(i/G,j/G)
#CP for informative prior
CPAlp=sum(BB1$`INC alpha`)/G
CPBet=sum(BB1$`INC Beta`)/G
CP=c(CPAlp,CPBet)
CP
#CP for 1/αβ
CPAlp1=sum(DD1$`INC alpha`)/G
CPBet1=sum(DD1$`INC Beta`)/G
CP1=c(CPAlp1,CPBet1)
CP1
#CP for 1/α
CPAlp2=sum(CC1$`INC alpha`)/G
CPBet2=sum(CC1$`INC Beta`)/G
CP2=c(CPAlp2,CPBet2)
CP2
```

Single sample Bayesian prediction using non-informative prior

Proposition I

i. When β is known, the following program was used:

```
Prob = function(lamtv, tau){
  d = ((1-(1+beta*T0)*exp(-beta*T0))*lamtv)/((beta^2)*tau*exp(-beta*tau))
  h = 0:(n-1)
  t = (d^h)/factorial(h)
  Z = t*exp(-d)
  ZN = sum(Z)
  Y = 1-ZN
  for(h in 0:(n-1)){
    return(Y)
  }
}
Prob(0.02, 500)
```

ii. When β is unknown, the following program was used:

```
st = sum(Data$Failure.Time) ## sum of ti's
g1 = function(b){
  (b^(2*n-1))*exp(-b*st)/((1-(1+b*T0)*exp(-b*T0))^n)
}
g2 = function (b, h){
  b1 = ((1-(1+b*T0)*exp(-b*T0))*lamtv)/((b^2)*tao*exp(-b*tau))
  b2 = (1-(1+b*T0))*exp(-b*T0)
  ((b1^h)/factorial(h))*exp(-b1)*g1(b)
}
H = 0:(n-1)
lamtv = 0.02
tao = 130
vec1 = rep(0,n)
for(j in 1:n){
  h = H[j]
  G = 1000
  vec2 = rep(0,G)
```

```
    vec3 = rep(0,G)
    for(i in 1:G){
      br = rgamma(1, 2*n,st)
      vec2[i] = g1(br)
      vec3[i] = g2(br,h)
    }
    vec1[j] = sum(vec3)/sum(vec2)
  }
  1-sum(vec1)
```

Proposition II

i. When β is known, the following program was used;

```
library(LambertW)
lamtv = 0.02
Chi=qchisq(0.10,df=2*n)
X = (1-(1+beta*T0)*exp(-beta*T0))
y = (-2*lamtv*X)/(beta*Chi)
GB = lambertWn(y)
TaoN = (-1/beta)*GB
Taostar = TaoN-T0
Taostar
```

Proposition III

i. When β is known, the following program was used;

```
tau = 130
DG = (((1-(1+beta*T0)*exp(-beta*T0))/((beta*T0*exp(-beta*T0) + exp(-
beta*T0)) - (beta*tau*exp(-beta*tau) + exp(-beta*tau))))^n)
DG
PAT <- function(j) {
  NUM <- (factorial(j-1)/((factorial(n-1))*(factorial(j-n))))*(((beta*T0*exp(-
beta*T0) + exp(-beta*T0)) - (beta*tau*exp(-beta*tau) + exp(-beta*tau)))^j)
  DEN <- (1 - (1 + beta*tau)*exp(-beta*tau))^j
```

```
  resultn1 <- NUM/DEN
  return(resultn1)
}

K_range <- 0:25
# Vector to store results
resultsn <- numeric(length(K_range))
for (i in 1:length(K_range)) {
  current_K <- K_range[i]
  total_sum <- 0
  for (j in n:(n + current_K)) {
    PAT1 <- PAT(j)
    total_sum <- total_sum + PAT1
  }
  resultsn[i] <- total_sum
}
resultsn
Prob = DG*resultsn
Prob
```

ii. When β is unknown:

```
V = function(b) {(b^(2*n-1))*exp(-b*sum(Data$Failure.Time))/((1-(1+b*T0)*exp(-b*T0))^n)}
c = integrate(V,0,Inf)
c$value

integrand <- function(b, j) {
  sum_term <- sum(Data$Failure.Time)
  g1 <- (((b*T0*exp(-b*T0)) + exp(-b*T0))-((b*tau*exp(-b*tau))+exp(-b*tau)))^(j-n)
  g2 <- ((1 - (1 + (b*tau))*exp(-b*tau)) ^ (j))
  g3 <- ((b^(2*n-1))*exp(-b*sum_term))
  result <- (gamma(j)/(gamma(j - n + 1)*gamma(n)))*g1*(1/g2)*g3
  return(result)
}

# Range for k
```

```
M <- 0:25
# Vector to store results
results1 <- numeric(length(M))
# Loop over each m and calculate the expression
for (i in 1:length(M)) {
  m <- M[i]
  total_sum <- 0

  for (j in n:(n + m)) {
    betint <- integrate(integrand, lower = 0, upper = Inf, j = j)$value
    total_sum <- total_sum + betint
  }

  results1[i] <- (1/c$value)*total_sum
}

results1
```

Proposition IV

i. When β is known, the following program was used;

```
tau = 700
Chi=qchisq(0.10,df=2*n)
X = (1-(1+beta*T0)*exp(-beta*T0))
UPL = (beta^2)*(tau*exp(-beta*tau))*Chi/(2*X)
UPL
```