A Covariate Software Reliability Model and Optimal Test Activity Allocation

Vidhyashree Nagaraju[1], Chathuri Jayasinghe[2], and Lance Fiondella[3]
[1]Tandy School of Computer Science, University of Tulsa, OK, USA
[2]Department of Statistics, University of Sri Jayewardenepura, Nugegoda, Sri Lanka
[3]Electrical and Computer Engineering, University of Massachusetts Dartmouth, MA, USA
Email: vidhyashree-nagaraju@utulsa.edu, chathuri.hettige@gmail.com, lfiondella@umassd.edu

***Abstract*—Traditional software reliability growth models enable quantitative assessment of the software testing process by characterizing fault detection in terms of testing time or effort. However, the majority of these models do not identify specific testing activities underlying fault discovery and thus can only provide limited guidance on how to incrementally allocate effort. Although there are several novel studies focused on covariate software reliability growth models, they are limited to model development, application, and assessment.**

This paper presents a non-homogeneous Poisson process software reliability growth model incorporating covariates based on the discrete Cox proportional hazards model. An efficient and stable expectation conditional maximization algorithm is applied to identify the model parameters. An optimal test activity allocation problem is formulated to maximize fault discovery. The proposed method is illustrated through numerical examples on a data set from the literature.

***Keywords*—software reliability growth model, non-homogeneous Poisson process, proportional hazards model, covariates, test activity allocation**

I. Introduction

Traditional software reliability growth models (SRGM) [1] characterize the fault discovery process during testing as a non-homogeneous Poisson process (NHPP). These models predict future faults as a function of testing time or effort, enabling inferences such as the number of faults remaining, additional time required to achieve a specified reliability, and optimization problems such as optimal release [2] and effort allocation [3]. However, the vast majority of these NHPP SRGM do not identify the underlying software testing activities that lead to fault discovery. Thus, effort allocation based on these models can only provide general guidance on the amount of effort to invest and limited information regarding the effectiveness of specific testing activities.

More recently, bivariate NHPP SRGM [4] and covariate models [5], which are capable of characterizing faults discovered as a function of multiple software testing activities such as calendar time, number of test cases executed, and test execution time have been proposed. Covariate models are an especially attractive alternative to testing effort models [6] because they introduce a single additional parameter per metric and do not require sequential model fitting procedures. However, covariate models do require stable and efficient numerical methods. To realize the full potential of covariate models, generalized optimization procedures such as test activity allocation are needed to guide the distribution of limited resources among specific test activities in order to maximize fault discovery, correction, and improved reliability.

Related research on covariate models includes the work of Rinsaka et al. [5] who combined the proportional hazards model and nonhomogeneous Poisson process to provide a generalized fault detection process possessing time-dependent covariate structure. Shibata et al. [7] subsequently extended this to a cumulative Bernoulli trial process. Okamura et al. [8] proposed a multi-factor software reliability model based on logistic regression and an efficient algorithm to estimate the model parameters. Okamura et al. [9] combined Poisson regression-based fault prediction and generalized linear models [10] with metrics-based software reliability growth models. Shibata et al. [11] implemented these methods in the M-SRAT (Metrics-based Software Reliability Assessment Tool).

Covariate models exhibit substantially improved prediction capabilities over NHPP software reliability growth model. Therefore, to enhance the utility of covariate models and encourage their use in practice, this paper makes the following primary contributions

- A software reliability growth model possessing a discrete Cox proportional hazard rate to incorporate covariates. This approach is analogous to the model introduced in [7], which is based on a discrete time model of Kalbfleisch and Prentice [12]. However, formulation and estimation of the proposed model ensures that the counts of software faults in disjoint intervals of the point process are independent random variables possessing a Poisson distribution with non-homogeneous rates, which is an assumption of a NHPP by design.
- A generalization of the testing effort allocation problem [13] to covariate models referred to as the *optimal testing activity allocation problem* to maximize fault discovery within a budget constraint.

The illustrations apply the model to a real data set from the literature with the expectation conditional maximization (ECM) algorithms and then solves the optimal testing activity allocation problem. The results indicate that periodic application of testing activity allocation could more effectively guide the type and amount of specific testing activities throughout the software testing process in order to discover more faults

despite limited resources.

The remainder of the paper is organized as follows: Section II discusses proportional hazards modeling incorporating covariates and formulates the optimal test activity allocation problem to maximize fault discovery. Section III describes model estimation, assessment, and selection. Section IV illustrates the proposed approach, while Section V provides conclusions and future research.

II. Proportional hazards modelling incorporating covariates

This section describes the formulation of a discrete Cox proportional hazards model incorporating covariates.

Testing occurs in discrete intervals $i = 1, 2, \ldots, n$. An NHPP software reliability model assumes that the point process possesses independent Poisson distributed increments. In other words, the number of faults detected over these n disjoint time intervals are independent. Suppose that we are interested in investigating the effect of p software test activities on fault detection. We denote the vector of software test activities in the ith testing interval by $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ for $i = 1, 2, \ldots, n$. The dependence of the functions on the parameters of the distribution of T and coefficients of the p software test activities in the model, denoted $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^{\mathbf{T}}$, must be estimated from data.

The Cox proportional hazards model for a discrete process [7] is

$$h_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} = 1 - (1 - h^0_{i;\theta})^{g(\mathbf{x_i};\boldsymbol{\beta})}, \tag{1}$$

for $i = 1, 2, \ldots, n$, where $h^0_{i;\theta}$ is known as the baseline hazard function and traditionally possesses the form

$$g(\mathbf{x_i};\boldsymbol{\beta}) = \exp(\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}). \tag{2}$$

Proof. From the definition of the discrete time hazard function

$$\begin{aligned} 1 - h_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} &= \frac{S_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}}{S_{i-1,\mathbf{x_{i-1}};\theta,\boldsymbol{\beta}}} \\ &= \left(\frac{S^0_{i;\theta}}{S^0_{i-1;\theta}}\right)^{g(\mathbf{x}_i;\boldsymbol{\beta})} \end{aligned} \tag{3}$$

However, $1 - h^0_{k;\theta} = \frac{S^0_{k;\theta}}{S^0_{k-1;\theta}}$. Therefore, Equation (3) is equal to

$$1 - h_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} = (1 - h^0_{i;\theta})^{g(\mathbf{x_i};\boldsymbol{\beta})} \tag{4}$$

and Equation (1) follows ■.

From Equation (4),

$$\begin{aligned} \prod_{k=1}^{n-1} (1 - h^0_{k;\theta})^{g(\mathbf{x_k};\boldsymbol{\beta})} &= \prod_{k=1}^{n-1} (1 - h_{k,\mathbf{x_k};\theta,\boldsymbol{\beta}}) \\ &= \prod_{k=1}^{n-1} \frac{S_{k,\mathbf{x_k};\theta,\boldsymbol{\beta}}}{S_{k-1,\mathbf{x_{k-1}};\theta,\boldsymbol{\beta}}} \\ &= S_{n-1,\mathbf{x_{n-1}};\theta,\boldsymbol{\beta}} \end{aligned} \tag{5}$$

so that

$$\begin{aligned} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} &= h_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} S_{i-1,\mathbf{x_{i-1}};\theta,\boldsymbol{\beta}} \\ &= \left(1 - (1 - h^0_{i;\theta})^{g(\mathbf{x_i};\beta)}\right) \prod_{k=1}^{i-1} (1 - h^0_{k;\theta})^{g(\mathbf{x_k};\boldsymbol{\beta})} \end{aligned} \tag{6}$$

follows from Equations (1) and (5).

The Cox model incorporates covariates $\mathbf{x}$ into the hazard rate and therefore assesses the effects they may have on fault detection. If the faults are assumed to be Poisson distributed, then a likelihood analysis can be performed using data available for estimation. Hence, it is possible to infer the effect of a covariate on fault detection despite the presence of multiple covariates.

Mean value function: The mean number of faults detected through the n^{th} interval is

$$H_{n;\omega,\theta,\boldsymbol{\beta}} = \omega \sum_{i=1}^{n} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} \tag{7}$$

A. Baseline hazard functions

The baseline hazard function in discrete time interval i possessing parameters θ is

$$h^0_{i;\theta} = \frac{p^0_{i,\theta}}{S^0_{i-1,\theta}} \tag{8}$$

where $p^0_{\cdot,\theta}$ and $S^0_{\cdot,\theta} = 1 - F^0_{\cdot,\theta}$ respectively denote the baseline probability mass function and survival function of the discrete time distribution T at the baseline levels of software test activities in the model.

1) Negative binomial of order two (NB): The negative binomial hazard rate

$$h^0_{i;b} = \frac{ib^2}{1 + b(i-1)} \tag{9}$$

is an example of hazard a function that can be substituted into Equation (6) to obtain various Cox proportional hazards models. Here b is the probability of detecting a fault and hence, $b \in (0, 1)$ and 2 indicates the order.

B. Optimal test activity allocation to maximize fault discovery

Similar to the concept of effort allocation [6] in NHPP software reliability growth models, it is possible to employ covariate models to guide resource allocation. Generalization of the testing effort concept to covariate models, requires division of resources across multiple activities, each of which possesses an effectiveness characterized by the parameter β_i but may also impose unique cost and time requirements. Examples associated with traditional software reliability testing include alternative black-box testing methods [14], with costs characterized by the hourly rates charged by skilled employees or consultants, whereas examples in the context of software security testing include traditional software reliability testing methods relevant to security as well as static and dynamic testing tools and techniques for exposing vulnerabilities.

Given n intervals of observed data and a budget of B resources to allocate to p activities (covariates), maximizing the total number of faults or vulnerabilities detected, so that they can be corrected prior to release is formulated as

$$\arg\max \widehat{H}_{(n+1);\omega,\theta,\boldsymbol{\beta}} \tag{10}$$

subject to

$$\sum_{j=1}^{p} \mathbf{c}_j \mathbf{x}_{j,(n+1)} \leq B$$

$$\mathbf{c}_j \left(\mathbf{x}_{j,(n+1)} - \mathbf{x}_{j,(n)} \right) = B_j \geq 0$$

where $\mathbf{c}_j > 0$ is the cost associated with an additional unit of activity j. In practice, the model can be fit to data from the first i intervals and Equation (10) solved based on the maximum likelihood estimates of the model and the budget to be allocated to activities during interval $(i+1)$.

III. Model Estimation, Assessment, and Selection

This section describes model estimation, assessment, and selection methods. Estimation methods include maximum likelihood estimation (MLE) and the expectation conditional maximization algorithm. Initial parameter estimation and measures of goodness of fit for model selection are also discussed.

A. Maximum likelihood estimation

For the purpose of estimation and inferential procedures, the likelihood function of the process needs to be constructed. The likelihood function is merely the joint distribution of the sample of observed values. In this case, the observed values are data: $(y_i, \mathbf{x_i}, i = 1, 2, \ldots, n)$, where n is the number of testing intervals. As mentioned in Section II, an NHPP software reliability model assumes that the point process possesses independent Poisson distributed increments. In other words, the number of faults detected over these disjoint time intervals are independent. Hence, the likelihood function of the NHPP SRGM given in Equation (6) is

$$\begin{aligned} \mathbf{L}(\theta, \beta, \omega) &= \Pr\left(Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n\right) \\ &= \prod_{i=1}^{n} \exp\left(-\omega p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}\right) \frac{\left(\omega p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}\right)^{y_i}}{y_i!} \\ &= \exp\left(-\omega \sum_{i=1}^{n} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}\right) \omega^{\sum_{i=1}^{n} y_i} \\ &\quad \prod_{i=1}^{n} \frac{p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}^{y_i}}{y_i!}, \end{aligned} \tag{11}$$

and the corresponding log-likelihood function is

$$\begin{aligned} LL(\theta, \boldsymbol{\beta}, \omega) &= -\omega \sum_{i=1}^{n} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}} + \sum_{i=1}^{n} y_i \ln(\omega) \\ &\quad + \sum_{i=1}^{n} y_i \ln(p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}) - \sum_{i=1}^{n} \ln(y_i!). \end{aligned} \tag{12}$$

incorporates the covariates $\mathbf{x}_i$ through $p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}$. The maximum likelihood estimates of the model parameters can be obtained by solving $\frac{\partial l}{\partial \omega} = 0$, $\frac{\partial l}{\partial \theta} = 0$, and $\frac{\partial l}{\partial \beta_j} = 0$, $j = 1, 2, \ldots, p$.

B. Expectation conditional maximization algorithm

This section describes the expectation conditional maximization [15], [16] algorithm to identify the maximum likelihood estimates of a model.

The steps to obtain the CM steps of the ECM algorithm of a covariate based NHPP SRGM are as follows:

- (S.1) Step one specifies the log-likelihood function as described in Equation (12).
- (S.2) Step two reduces the log-likelihood function from ν to $(\nu - 1)$ parameters by differentiating the log-likelihood function with respect to ω, equating the result to zero, and solving for ω to produce

$$\widehat{\omega} = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} p_{i,\mathbf{x_i};\theta,\boldsymbol{\beta}}}, \tag{13}$$

  which is then substituted into the log-likelihood function to obtain the reduced log-likelihood (RLL) function.
- (S.3) Step three derives the conditional maximization steps for the remaining $(\nu - 1)$ parameters by computing partial derivatives

$$\frac{\partial RLL}{\partial \theta} = 0 \tag{14}$$

  and

$$\frac{\partial RLL}{\partial \boldsymbol{\beta}} = 0 \tag{15}$$

  where θ denotes all parameters except ω and $\boldsymbol{\beta}$ are the coefficients of the p covariates.
- (S.4) Step four cycles through the $(\nu - 1)$ CM-steps holding the other $(\nu - 2)$ parameters constant, applying a numerical root finding algorithm to determine the maximum likelihood estimates $\widehat{\theta}/\omega$. This process continues until a user specified convergence criterion is achieved.
- (S.5) Step five computes the MLE of ω by substituting the estimates of θ and $\boldsymbol{\beta}$ into Equation (13), producing the MLE for all ν parameters of the model.

Steps (S.1) through (S.5) can be applied to different hazard functions introduced in Section II-A. Step (S.4) reduces a $(\nu - 1)$-dimensional problem to $(\nu - 1)$ single-dimensional problems, enabling the application of a stable numerical method in each CM-step. Coupled with the monotonicity of the ECM algorithm, this ensures convergence to the maximum likelihood, which is especially important in an open source implementation [17].

C. Initial parameter estimation

The initial parameter estimates [18] of parameter ω is

$$\omega^{(0)} = n \tag{16}$$

The remaining parameters of the distribution function $F(\bullet; \theta)$, which corresponds to the term $\sum_{i=1}^{n} p_{i,\boldsymbol{x}_i;\theta,\boldsymbol{\beta}}$ in Equation (7) are computed as

$$\theta^{(0)} := \sum_{i=1}^{n} \frac{\partial}{\partial \theta} \log\left[f(;\theta,\boldsymbol{\beta})\right] = 0. \tag{17}$$

D. Goodness of fit measures

This section summarizes some goodness of fit measures to assess how well a model characterizes a failure data set.

1) Akaike Information Criterion: The Akaike Information Criterion [19] is an information theoretic measure of a model's goodness of fit. The AIC quantifies the tradeoff between model precision and complexity. The AIC of model i is a function of the maximized log-likelihood and the number of model parameters (ν).

$$AIC_i = 2\nu - 2LL(\mathbf{x_i}; \widehat{\omega}, \widehat{\theta}, \widehat{\beta}) \tag{18}$$

The term 2ν of Equation (18) is a linearly increasing penalty function for the number of parameters, while $LL(\mathbf{x_i}; \widehat{\omega}, \widehat{\theta}, \widehat{\beta})$ is the log-likelihood function of failure data with covariates $\mathbf{x_i}$ evaluated at the maximum likelihood estimate. Model j preserves information better than model i and is preferred with statistical significance if $AIC_{i,j} = AIC_i - AIC_j > 2.0$ [20].

2) Bayesian Information Criterion: The Bayesian information criterion of model i is a function of the maximized log-likelihood, number of model parameters ν, and the sample size n

$$BIC_i = -2LL(\mathbf{x_i}; \widehat{\omega}, \widehat{\theta}, \widehat{\beta}) + \nu \log(n) \tag{19}$$

The penalty term of the BIC is therefore proportional to the number of parameters ν multiplied by the logarithm of the sample size n.

3) Sum of Squares Error (SSE): The sum of squares error, also known as the residual sum of squares, for failure count data is

$$SSE = \sum_{i=1}^{n} (\widehat{H}_{i;\omega,\theta,\beta} - Y_i)^2 \tag{20}$$

where $Y_i = \sum_{j=1}^{i} y_j$ is the cumulative number of faults observed in the first i time intervals.

4) Predictive Sum of Squares Error (PSSE): PSSE compares the predictions of a model with data not used to perform model fitting. The predictive sum of squares error for failure count data is

$$PSSE = \sum_{i=n-\ell+1}^{n} (\widehat{H}_{i;\omega,\theta,\beta} - Y_i)^2 \tag{21}$$

where the maximum likelihood estimates of the model parameters are determined from the first $n - \ell$ intervals.

IV. Illustrations

This section demonstrates the model and optimal covariate allocation. The DS1 data set [7] is employed, which contains $n = 17$ weeks of observations. The data set possess three covariates, including execution time (E) in hours, failure identification work (F) in person hours, and computer time failure identification (C) in hours.

The first example compares goodness of fit measures attained by the negative binomial model on all possible subsets of covariates contained in the DS1 data set. The second example assesses optimal effort allocation.

A. Goodness of fit model assessment

This section systematically compares the models with negative binomial hazard function considering all eight possible combinations of three covariates in data set DS1.

Table I summarizes the goodness of fit of the negative binomial hazard rate function for each possible combination of covariates possessing ν parameters. Preferred combinations of covariates with respect to the LLF, AIC, BIC, SSE, and PSSE are indicated in bold. While the model fit using all three covariates achieves the highest log-likelihood value, only SSE prefers EFC on DS1, whereas the F covariate is preferred by AIC and BIC, while PSSE suggests EF. The SSE and PSSE provide conflicting results. Specifically, SSE prefers all three covariates one DS1, whereas PSSE is lowest for a model with EF covariates. However, some combinations of one or more covariate also attain a relatively low SSE and PSSE, suggesting that practical model selection [21] requires a tradeoff between information theoretic and predictive measures of goodness of fit.

TABLE I: Goodness of fit measures for negative binomial hazard function model on DS1

β	ν	DS1				
		LLF	AIC	BIC	SSE	PSSE
-	2	-36.41	76.82	78.49	208.35	3.99
E	3	-32.31	70.62	73.12	70.16	3.16
F	3	-28.80	**63.60**	**66.10**	25.94	1.16
C	3	-31.96	69.91	72.41	81.73	11.12
EF	4	-28.05	64.11	67.44	18.33	**0.96**
FC	4	-28.37	64.75	68.08	20.77	2.19
EC	4	-28.97	65.95	69.28	33.02	10.18
EFC	5	$\mathbf{-27.29}$	64.58	68.74	**11.89**	2.44

B. Optimal test activity allocation

This section illustrates optimal test activity allocation formulated in Section II-B in the context of the negative binomial model on DS1. In practice, optimal test activity allocation can be applied to the model with the subset of covariates determined by the user based on their subjective preference for one or more measures of goodness of fit. For the sake of illustration, the negative binomial model was fit to each combination of covariates with the first $n - 1$ intervals. The amount of effort originally allocated in the nth interval of DS1 was $E = 7.6$, $F = 24$, and $C = 8$. Thus, we optimally allocate a budget of $B = 39.6$ resources to each subset of covariates, according to the fitted models. For simplicity, each covariate was assumed to possess unit cost ($c_i = 1.0$), but the formulation in Section II-B is capable of considering non uniform costs.

Table II lists each combination of one or more covariate, the estimated number of faults that would occur with optimal allocation ($\widehat{H}^*_{n;\omega,\theta,\beta}$) using Equation (10), and the percent of the budget allocated to the available covariates. Table II indicates that optimal allocation of effort to the model with all three covariates dedicates 38.03% of the budget to activity E and the remaining 61.97% to activity C, which respectively correspond to 15.06 and 24.54 units of the original budget.

The estimated number of faults is 1.55, whereas allocating effort according to the nth interval of DS1 ($E = 7.6$, $F = 24$, and $C = 8$) estimates only 0.92 faults. These results suggest that optimal allocation could expose nearly 1.7 times as many faults, indicating more efficient allocation could have conserved significant resources. For the sake of comparison, we increased the original allocations reported in the nth interval by a common factor greater than one until the number of faults estimated was equal to the optimal allocation. That is, each of the three covariates was multiplied by the same factor until 1.55 faults were achieved. It was determined that a total budget of 79.2 would be required to identify the same number of faults with this allocation strategy, requiring two times as much effort.

TABLE II: Optimal covariate effort allocation to DS1 with negative binomial hazard function

β	$\widehat{H}^*_{n;\omega,\theta,\beta}$	$\%E$	$\%F$	$\%C$
E	6.63	100	–	–
F	2.61	–	100	–
C	0.41	–	–	100
EF	5.30	100	–	–
FC	1.97	–	5.70	94.30
EC	0.62	40.75	–	59.25
EFC	1.55	38.03	–	61.97

V. Conclusions and future research

This paper presents a covariate software reliability growth model and formulates the optimal test activity allocation problem to maximize fault discovery. Expectation conditional maximization algorithms were derived and applied to a data set from the literature. The optimal test activity allocation problem was then solved to illustrate how it could efficiently increase the number of faults discovered by distributing limited testing resources among the testing activities performed.

Future research will develop additional optimization problems for covariate models to guide activities during the testing process.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Number (#1749635) the Army Research Laboratory under Cooperative Agreement Number W911NF-19-2-0224. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] W. Farr and O. Smith, "Statistical modeling and estimation of reliability functions for software (SMERFS) users guide," Naval Surface Warfare Center, Dahlgren, VA, Tech. Rep. NAVSWC TR-84-373, Rev. 2, 1984.

[2] M. Zhao and M. Xie, "Robustness of optimum software release policies," in *IEEE International Symposium on Software Reliability Engineering*, 1993, pp. 218–225.

[3] L. Fiondella and S. Gokhale, "Software reliability models incorporating testing effort," *Opsearch*, vol. 45, no. 4, pp. 351–368, 2008.

[4] T. Ishii, T. Fujiwara, and T. Dohi, "Bivariate extension of software reliability modeling with number of test cases," *Intl. Journal of Reliability, Quality and Safety Engineering*, vol. 15, no. 1, pp. 1–17, 2008.

[5] K. Rinsaka, K. Shibata, and T. Dohi, "Proportional intensity-based software reliability modeling with time-dependent metrics," in *IEEE International Conference on Computer Software and Applications*, vol. 1, 2006, pp. 369–376.

[6] S. Yamada, H. Ohtera, and H. Narihisa, "Software reliability growth models with testing-effort," *IEEE Transactions on Reliability*, vol. 35, no. 1, pp. 19–23, 1986.

[7] K. Shibata, K. Rinsaka, and T. Dohi, "Metrics-based software reliability models using non-homogeneous Poisson processes," in *IEEE International Symposium on Software Reliability Engineering*, 2006, pp. 52–61.

[8] H. Okamura, Y. Etani, and T. Dohi, "A multi-factor software reliability model based on logistic regression," in *IEEE International Symposium on Software Reliability Engineering*, 2010, pp. 31–40.

[9] H. Okamura and T. Dohi, "A novel framework of software reliability evaluation with software reliability growth models and software metrics," in *IEEE International Symposium on High-Assurance Systems Engineering*, 2014, pp. 97–104.

[10] ——, "Towards comprehensive software reliability evaluation in open source software," in *IEEE International Symposium on Software Reliability Engineering*, 2015, pp. 121–129.

[11] K. Shibata, K. Rinsaka, and T. Dohi, "M-SRAT: Metrics-based software reliability assessment tool." *International Journal of Performability Engineering*, vol. 11, no. 4, 2015.

[12] J. Kalbfleisch and R. Prentice, "Marginal likelihoods based on Cox's regression and life model," *Biometrika*, vol. 60, no. 2, pp. 267–278, 1973.

[13] S. Yamada, J. Hishitani, and S. Osaki, "Software-reliability growth with a weibull test-effort: A model and application," *IEEE Transactions on Reliability*, vol. 42, no. 1, pp. 100–106, 1993.

[14] P. Ammann and J. Offutt, *Introduction to software testing*. Cambridge University Press, 2016.

[15] V. Nagaraju, L. Fiondella, P. Zeephongsekul, C. Jayasinghe, and T. Wandji, "Performance optimized expectation conditional maximization algorithms for nonhomogeneous Poisson process software reliability models," *IEEE Transactions on Reliability*, vol. 66, no. 3, pp. 722–734, 2017.

[16] P. Zeephongsekul, C. Jayasinghe, L. Fiondella, and V. Nagaraju, "Maximum-likelihood estimation of parameters of NHPP software reliability models using expectation conditional maximization algorithm," *IEEE Transactions on Reliability*, vol. 65, no. 3, pp. 1571–1583, 2016.

[17] V. Nagaraju, V. Shekar, J. Steakelum, M. Luperon, L. Fiondella, and Y. Shi, "Practical software reliability engineering with the software failure and reliability assessment tool (sfrat)," *SoftwareX*, vol. 10, p. 100357, 2019.

[18] H. Okamura, Y. Watanabe, and T. Dohi, "An iterative scheme for maximum likelihood estimation in software reliability modeling," in *IEEE International Symposium on Software Reliability Engineering*, Nov 2003, pp. 246–256.

[19] L. Fiondella and S. Gokhale, "Software reliability model with bathtub-shaped fault detection rate," in *Annual Reliability and Maintainability Symposium*, 2011, pp. Session 9D–2.

[20] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, "Akaike information criterion statistics," *Dordrecht, The Netherlands: D. Reidel*, p. 81, 1986.

[21] K. Sharma, R. Garg, C. Nagpal, and R. Garg, "Selection of optimal software reliability growth models using a distance based approach," *IEEE Transactions on Reliability*, vol. 59, no. 2, pp. 266–276, 2010.