JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 343-355 (2022)
DOI: 10.6688/JISE.202203_38(2).0004

Testing-Effort based NHPP Software Reliability Growth Model with Change-point Approach

VISHAL PRADHAN+, JOYDIP DHAR AND AJAY KUMAR
Department of Applied Sciences
ABV-Indian Institute of Information Technology and Management
Gwalior, 474015 India
E-mail: {vishalp; jdhar; ajayfma}@iiitm.ac.in

A software project managers can execute well-prepared research tasks to utilize associated cost-effectively testing resources using software reliability growth models (SRGMs). Over the last four decades, several SRGMs are introduced to estimate reliability growth and applicable particular to software development research. So far, it seems that very few numbers of SRGMs recognize potential adjustments in test-effort consumption. In certain instances, testing-resource allocation practices may be modified with time. Thus, this study integrates the essential principle of multiple change-points with the testing-effort function in proposed models. Two benchmark datasets illustrate the efficiency and applicability of the proposed models. Normalized criteria distance is used to evaluate the models ranking based on four comparison criteria on two failure datasets. Experimental outcomes show that the proposed models offer reasonably better fault predictability compare to other models.

Keywords: non-homogeneous poisson process (NHPP), reliability model, software reliability growth models (SGRMs), perfect debugging, change-point, testing-effort

1. INTRODUCTION

Software development is a time-consuming and costly process due to the implementation complexity, deadline constraint and testing resources. Therefore, the key priority of software industry's is to reduce the development cost to an appropriate level and improves the software systems efficiency. It should be remembered that software testing costs are extremely high. Practically, it can surpass half the total software development expense [1]. It is challenging to deliver a stable and functional software product on schedule and within budget, because of its complexity. Therefore, project managers have to address many management and technical problems during software development, such as frequent failure rate, low-quality, cost overrun and delayed delivery. This work primarily focuses on the development of software reliability growth models (SRGMs) to examine the initial number of faults, fault detection rate, and software reliability.

In literature, stochastic-driven non-homogeneous Poisson process (NHPP) based SRGMs are developed to evaluate software reliability growth. SRGMs fore-casted soft-

Received October 29, 2020; revised Dec. 3, 2020 & Jan. 20 & Feb. 2, 2021; accepted March 1, 2021.
Communicated by Chu-Ti Lin.
+ Corresponding author: vishal.iiitmg@gmail.com.

ware reliability and addressed the variations that arise throughout the developing process. During the software development activities, SRGM offers essential information for decision-making, such as cost analysis. It indicates how many failures customers will ultimately experience and determine the growth of the testing process [2]. Future software system failure behavior is predetermined by modelling and examining its prior failure pattern. Ensuring the consistency of the systems, software execute their functions correctly is more important. Two major factors affect reliability among various SRGMs: detection rate of faults and the initial amount of faults. Software quality and reliability modelling are important as the software is practiced in different applications.

In the literature, numerous software reliability estimation and prediction models are proposed to assess and measure the reliability growth [3–6]. Several researchers presume a constant fault detection rate to derive their SRGMs. They believe that all faults are equally likely to be found during testing, and the detection rate continues constant across the periods. They have a general assumption that the software detection rate is the same throughout the testing phase. In fact, the fault detection rate (FDR) depends heavily on programme size, test team skills, and testability. Hence, it can be changeable, and this method will clearly boost software efficiency and testing. Some factors in the realistic development phase are also considered to boost the efficiency of SRGMs, *e.g.*, testing coverage, fault reduction factor, time-delay correction, testing effort (TE), change-point (CP), *etc*. [7–10].

Developers and project managers must adequately allocate limited resources, personnel, production, and TE during the development process [11]. Therefore, programmers are struggled to build an error-free software system due to diverse interface and logic design errors. Some studies discussed the associations among reliability and TE expenditure for growth analysis [12–14]. Kapur *et al*. [15] studied the TE-dependent learning process and classified the faults into two types based on TE's need to correct them. Practically, TE may not be smooth and can be changed during the testing process. Further, few studies considered the CP concept in their SRGMs as the TE expenditure may not be constant over time [16]. Lin and Huang [17] considered multiple CPs into the flexible Weibull-type time-dependent TE, which seems to be more realistic. Lin and Huang [17], suggested that the multiple CPs should be incorporated due to the changing TEs in reality.

The motivation and key emphasis of the above studies is to introduce a software reliability modelling approach that incorporates TE with CP factor. The rest of the paper is organized as follows. Section 2 presents the background and a brief discussion on the related work. Section 3 provides the formulation of the proposed models. Section 4 presents experimental results along with the description of datasets, parameter estimation, and evaluation criteria. Finally, Section 5 concludes the paper with future work.

Acronyms and notations used in this paper are listed in Table 1 .

2. BACKGROUND AND RELATED WORK

The NHPP provides the mathematical formulation for describing the software failure phenomenon to model SRGMs. The primary challenge in the NHPP based SRGMs is to determine an appropriate mean value function (MVF). The pioneer NHPP based Goel and Okumoto (G-O) model [18] to determine the MVF is given in the following section.

Table 1. Acronyms and notations along with descriptions.

Acronyms	Descriptions	Notations	Descriptions
SRGM	software reliability growth model	$\chi(t)$	total number of initial faults present
NHPP	non-homogeneous Poissin process	$\psi(t)$	fault detection rate per unit time
MVF	mean value function	$s(t)$	expected number of faults detected by t
TE	testing-effort	$G(t)$, $g(t)$	cumulative and current TE, respectively
CP	change-point	$\mathscr{T}$	change-point
NCD	normalized distance criteria	ρ	TE scaling factor
FDR	fault detection rate	$R(t)$	reliability by time t

2.1 Basic Terminology

The representation for the observed amount of faults is $\mathscr{N}(t)$, whose MVF is $s(t)$, and formulation of NHPP-based SRGM is as follows:
$\{\mathscr{N}(t),(t \geq 0)\}$ expresses the total number of faults observed up to execution time t. During the software testing, defect/failure follows the NHPP.

$$Pr\{\mathscr{N}(t)=\omega\}=\frac{(s(t))^{\omega}e^{-s(t)}}{\omega!}, \qquad \omega=1,2,3,\ldots \tag{1}$$

where, $s(t)$ is a expected number of detected faults by time t, ω is failures occurring by time t. The intensity function $(\vartheta(t))$ and MVF $(s(t))$ is represented as:

$$\vartheta(t)=\frac{ds(t)}{dt} \qquad or \qquad s(t)=E[\mathscr{N}(t)]=\int_0^t \vartheta(x)dx. \tag{2}$$

In terms of initial faults, MVF $(s(t))$ of NHPP based SRGM is given as follows:

$$\frac{ds(t)}{dt}=\psi\times[\chi-s(t)]. \tag{3}$$

Here, ψ and χ are FDR and total faults present in software, respectively. This most popular NHPP based SRGM is widely known as G-O model.

Testing effort (TE): It is defined as efforts needed to correct the faults during the testing time, which play a crucial role in reliability estimation. The TE is taken into consideration as an essential resource expenditure, covered human resources, number of test cases, CPU time, *etc*. Due to the allocation policy effect, it may not be acceptable to ignore TE's consumption rate variance. Pachauri *et al*. [19] considered potential changes in the TE consumption rate and modeled them by generalized Weibull distribution. The relation between cumulative and current TE is represented as:

$$G(t)=\int_0^t g(t)dt \quad or \quad \frac{dG(t)}{dt}=g(t). \tag{4}$$

Here, cumulative TE $(G(t))$ and current TE $(g(t))$ expenditure.

Change Point (CP): It is an influential factor that helps to examine the changing scenarios during the testing phase [20]. It happens due to changes in the testing strategy, testing

Table 2. A summary of existing SRGMs, model types, and MVF.

SRGMs	Model type	MVF (*s(t)*)	Description
G-O model [18]	Concave *(C)*	$\chi(1-e^{-\psi t})$	Exponential model with constant FDR
DSS model [29]	S-shaped *(S)*	$\chi(1-(1+\psi t)e^{-\psi t})$	GO model changed time dependent FDR
ISS model [30]	S-shaped *(S)*	$\frac{\chi\left(1-e^{-\psi t}\right)}{1+\xi e^{-\psi t}}$	For ξ = 0 its become GO model
Y-IFD model [31]	Concave *(C)*	$\chi\left(\left[1-e^{-\psi t}\right]\left[1-\frac{\zeta}{\psi}\right]+\zeta t\right)$	Assume ζ as a fault introduction rate (FIR)
Pham-Zhang model [1]	Concave *(C)*	$\chi\left[1-e^{-\psi t}\right]\left[1+(\psi+\zeta)t+\psi\zeta t^2\right]$	Incorporate ζ (FIR) and combine with FDR
PNZ model [32]	Concave *(C)*	$\frac{\chi\left(\left[1-e^{-\psi t}\right]\left[1-\frac{\zeta}{\psi}\right]+\zeta t\right)}{1+\xi e^{-\psi t}}$	Consider linear function of ζ (FIR) with ISS FDR
C-TC model [33]	S-shaped *(S)*	$\chi\left[1-\left(\frac{\xi}{\xi+(\psi t)^{\zeta}}\right)^{\alpha}\right]$	Model with testing coverage function

environment, fault density, testing effort, and so on. Therefore, FDR may not be smooth and may change at some point called CP and represented by $\mathscr{T}$ [16]. Zhao [21] claims that if the organization's testing methods and resource allocations change, a CP tends to occur before the programme is released for field activity.

2.2 Related Work

Chiu *et al*. [22] have suggested TE-dependent improvement and learning TE function over time. Chang [23] introduced NHPP based SRGMs with CPs, which shows good performance in reliability growth modelling. In 2016, Chatterjee and Shukla [24] suggested an SRGM considering imperfect debugging, fault dependency, and CP. They also analyzed the CPs effect on various parameters, such as rate of fault introduction and fault removal rate. Ahmad *et al*. [25] introduced the model that considers exponentiated Weibull TE to develop a NHPP based ISS-SRGM with an imperfect debugging environment. Shyur [26] also found SRGMs with both issues, *i.e*., CP and imperfect debugging. In 2019, Chatterjee and Shukla [27] suggested a method to model SRGM with imperfect debugging, test coverage, and CP. Lin and Huang [17] demonstrated how to integrate multiple CP assumptions into the Weibull TE function and formulate NHPP based SRGM. In 2011 Huang *et al*. proposed generalized multiple CP for reliability estimation [28]. The above-discussed models consider either TE or CP or both as attributes to overcome the existing problems. However, some models either considered the changes in TE with scaling factor, or they do not believe in the CP attribute.

Change-point (CP) is a useful attribute to study the changing scenarios during the testing phase. It happens due to change in testing strategy, environment, and effort, changes in detection rate is possible. CPs can affect the execution of TE and reliability evaluations throughout the testing phase. Therefore, this study integrates both TE function and CP factor to model a robust SRGMs. In the proposed model, we consider the changes in TE function by incorporating scaling factor in it. TE function is modeled through a testing time-dependent power function. We also consider the changes in FDR. This hypothesis makes the SRGM more flexible, and it can capture the variation of reliability growth curves. By doing so, the detection rate becomes more realistic.

In Table 2, existing SRGMs, model types, their MVFs and descriptions are given. SRGMs mentioned in Table 2 are used for the comparison with proposed SRGMs.

3. SOFTWARE RELIABILITY GROWTH MODELING

This section shows how to incorporate the TE function and CP into the SRGM. The time-dependent current TE ($g(t)$) and cumulative TE ($G(t)$) expenditure by time $(0,t]$ can be formulated as a power function of the testing time t. Customarily, it grows very fast from the beginning of the testing. As more test cases are performed to investigate the software, the TE growing rate becomes flat and less because less TE is needed to correct the residual faults after some certain time point.

$$g(t) = W\rho \times t^{\rho-1} \qquad or \qquad G(t) = W.t^{\rho} \qquad \rho \geq 0 \tag{5}$$

In the proposed model, the CP factor is incorporated. Therefore, we apply TE scaling factor by the parameter p, which represents the TE fraction and defined as:

- For single CP

$$G^{\mathscr{T}}(t) = \begin{cases} p \times G(t) & 0 \leq t \leq \mathscr{T} \\ (1-p) \times G(t) & t > \mathscr{T} \end{cases} \tag{6}$$

- For multiple CP

$$G^{\mathscr{T}}(t) = \begin{cases} p_1 \times G(t) & 0 \leq t \leq \mathscr{T}_1 \\ p_2 \times G(t) & \mathscr{T}_1 < t \leq \mathscr{T}_2 \\ \dots & \dots \\ p_n \times G(t) & t > \mathscr{T}_{n-1} \end{cases} \tag{7}$$

where, $\sum_{i=1}^{n} p_i = 1$ for n = 1, 2, 3, ..., and $\mathscr{T}_1$, $\mathscr{T}_2$, ... $\mathscr{T}_n$ represent CPs.

For software reliability modelling, the following assumptions are made [13, 18, 34]:

1. The fault identification or the counting process follows the NHPP.
2. Due to the existing faults, the software systems are subject to failure at random.
3. All the faults in a program are different in nature and can be assumed as statistically independent.
4. The mean number of detected faults in the time interval $(t, t+\Delta t]$ by the current TE expenditures is proportional to the expected number of faults present in the system.
5. The time-dependent behavior of TE can be modeled by the power function of testing time t with CP(s).

3.1 SRGM Without CP (Proposed Model-1)

According to the assumption of TE, the MVF is obtained by solving the following differential equation:

$$\frac{ds(t)}{dt} \times \frac{1}{g(t)} = \psi(t) \times [\chi(t) - s(t)], \tag{8}$$

$$s(t)=\chi\times\left[1-e^{-\psi\times\int_0^t g(z)dz}\right]=\chi\times\left[1-e^{-\psi\times(G(t)-G(0))}\right], \tag{9}$$

$$s(t)=\chi\times\left[1-e^{-\psi\times G^*(t)}\right]=\chi\times\left[1-e^{-\psi\times p\times Wt^\rho}\right]. \tag{10}$$

Where, $\psi(t)$ and $\chi(t)$ are detection rate and total faults present in the software, respectively.

3.2 SRGM With Single CP (Proposed Model-2)

In this model, we incorporate a single CP. The FDR and TE changes over time due to change in the environmental factors and resource allocation. Hence, the fault detection rate is defined as follows:

$$\psi(t)=\begin{cases}\psi_1 & 0\le t\le \mathscr{T}\\ \psi_2 & t\ge \mathscr{T}\end{cases}. \tag{11}$$

The detection rate before and after CP is ψ_1 and ψ_2, respectively. Now incorporating the effect of CP on TE by Eq. (6) and the single CP model is expressed as follows:

$$s(t)=\begin{cases}\chi\times\left[1-e^{-\psi\times p\times Wt^\rho}\right] & 0\le t\le \mathscr{T}\\ \chi\times\left[1-e^{-(\psi_1\times p\times W(\mathscr{T}^\rho+\psi_2\times(1-p)\times(t^\rho-\mathscr{T}^\rho))}\right] & t>\mathscr{T}\end{cases}. \tag{12}$$

Both the models are proposed to address the problem of failure behavior. In the first model, we incorporate the testing effort factor to forecast the number of failures observed by time t. The second model incorporates an important factor change point that is useful to study the changing scenarios during the testing phase. Similarly, generalized n-CP can also be incorporated in the proposed model. The MVF is expressed as follows:

$$s(t)=\begin{cases}\chi\times\left[1-e^{-\psi\times p_1\times Wt^\rho}\right] & 0\le t\le \mathscr{T}_1\\ \chi\times\left[1-e^{-(\psi_1\times p_1\times W\mathscr{T}_1^\rho+\psi_2\times p_2\times W(t^\rho-\mathscr{T}_2^\rho))}\right] & \mathscr{T}_1<t\le \mathscr{T}_2\\ \cdots & \cdots\\ \chi\times\left[1-e^{-\left(\sum_{i=1}^n \psi_i\times p_i\times W(\mathscr{T}_n^\rho-\mathscr{T}_{n-1}^\rho)\right)}\right] & t>\mathscr{T}_{n-1}\end{cases}. \tag{13}$$

In the next section, we estimate the parameters and examine the experimental results.

4. EXPERIMENTS

This section elaborates on the datasets descriptions, evaluation criteria, ranking method and experimental results to evaluate the effectiveness of proposed models over the existing competing models.

Table 3. Model evaluation criteria.

S.No.	Criteria	Formula	Measure
1	MSE	$\frac{\sum_{r=1}^{k}(s(t_r)-\gamma_r)^2}{k-\theta}$	Measure distance between observed and model value
2	PRR	$\sum_{r=1}^{k}\left(\frac{(s(t_r)-\gamma_r)}{s(t_r)}\right)^2$	Distance between observed and model value against model value
3	PP	$\sum_{r=1}^{k}\left(\frac{(s(t_r)-\gamma_r)}{\gamma_r}\right)^2$	Distance between observed and model value against observed value
4	Variance	$\sqrt{\frac{1}{k-1}\sum_{r=1}^{k}(\gamma_r-s(t_r)-Bias)^2}$	Bias = $\frac{\sum_{r=1}^{k}(s(t_r)-\gamma_r)^2}{k}$. (model indicates fit for small variance)

4.1 Datasets

We test the existing and proposed NHPP based SRGMs on two real life benchmark datasets. We have taken DS-1 and DS-2 due to its variability. Besides, other datasets had some uniform type behavior. By taking a diverse datasets, we show how good the models will behave.

DS-1: This dataset is originally taken from a report of the real-time control system [35]. The monitoring programme has about 200 modules and each module with an average of 1000 high-level language lines. This data reveals the cumulative failure observed during the 111-day test cycle is 481. For this data, CP ($\mathscr{T}$) is 35.

DS-2: This dataset is obtained and organized on Firefox from Bugzilla [2]. For this data, CP ($\mathscr{T}$) is 50, where 56 software failures between test start and CP while 54 software failures between CP and the end of the test. This data reveals the cumulative failure observed during the 81-day test cycle is 116.

4.2 Evaluation Criteria and Model Ranking

The four most prominent evaluating criteria, *e.g.*, mean square error (MSE), predictive-ratio risk (PRR), predictive power (PP) and variance, are used to judge the efficacy of the proposed models. The expression of these evaluating criteria with descriptions are given in Table 3.

Additionally, we use the *normalized criteria distance (NCD)* [36] measure used for ranking and selecting the best model from the competing models based on the aforementioned evaluating criteria. The expression of *NCD* is given as follows:

$$NCD_j = \sqrt{\sum_{m=1}^{d}\left(\frac{\mathscr{C}_{jm}}{\sum_{n=1}^{s}\mathscr{C}_{nm}}\right)^2 \mathscr{W}_m} \qquad m = 1,2,\dots,d \qquad n = 1,2,\dots,s. \tag{14}$$

Here, s and d are the number of models and number of evaluation criteria, respectively. $\mathscr{W}_m$ represents weight and $\mathscr{C}_{nm}$ denotes jth model criterion value. Here the model with a smaller NCD value is ranked as more reliable or best-suited model for failure prediction.

When the value of relative errors (REs) for the software reliability model is close to zero, we can say that it supports its ability to provide a more precise forecast. Predictive

Table 4. Estimated value of parameters and SSE for DS-1 and DS-2.

DS	Model	χ	ψ/ψ_1	ρ/ξ	ψ_2/ζ	p/α	W	SSE
DS-1	G-O	497.29	0.031	-	-	-	-	109662.72
	DSS	488.40	0.066	-	-	-	-	37517.800
	ISS	482.02	0.070	4.146	-	-	-	32531.760
	Yamada-IFD	591.80	0.024	0.002	-	-	-	648074 .52
	Pham-Zhang-IFD	482.00	0.081	0.007	-	-	-	48616.200
	PNZ model	470.76	0.075	0.0002	4.693	-	-	34282.800
	C-TC model	483.70	0.02975	116.30	1.476	12.35	-	32679.735
	Proposed Model-1	483.9654	0.005381	1.501601	-	1.00	2.301	32255.172
	Proposed Model-2	476.4088	0.062038	1.209668	0.041497	0.213	2.710	17624.398
DS-2	G-O	170.08	0.012	-	-	-	-	7733.310
	DSS	280.34	0.063	-	-	-	-	14640.28
	ISS	195.01	0.015	0.804	-	-	-	7902.180
	Yamada-IFD	129.56	0.0144	0.0053	-	-	-	7137.000
	Pham-Zhang-IFD	120.00	0.045	1×10^{-6}	-	-	-	14635.14
	PNZ model	120.06	0.021	0.005	0.423	-	-	7722.330
	C-TC model	146.00	0.7746	4.423	0.7046	0.2717	-	13117.87
	Proposed Model-1	198.001	0.015164	0.890799	-	1.00	1.045	7075.068
	Proposed Model-2	150.003	0.284592	0.455754	0.996641	0.306	1.702	323.4462

validity, which can be expressed by computing the RE for a dataset. RE is the ability of the model to predict the present and future failure behavior from the past failure data. The RE used to compare the different models is defined as follows:

$$Relative\ \ Error\ \ (RE) = \frac{[s(t_r) - s_r]}{s_r}. \tag{15}$$

4.3 Parameter Estimation

After the mathematical formulation of $s(t)$, its parameters are usually estimated either with the least square estimation (LSE) or maximum likelihood estimation (MLE). Here, in this work, we employ LSE for parameter estimation and compute the sum of square error (SSE) for analyzing SRGM forecast results. The SSE and estimated parameter values of considered models for DS-1 and DS-2 are shown in Table 4. Table 4 also gives a close look for a few other pre-existing SRGMs estimated value of parameters. Here, it is observed that the SSE values for the proposed models are far better than the competing models for both datasets.

Moreover, from the parameter estimates given by the proposed model, some additional details can be obtained. The initial fault content for DS-1 is estimated 483.9654 and 476.4088 by proposed models 1 and 2, respectively. Also, the initial fault content for DS-2 is estimated as 198.001 and 150.003 by proposed models 1 and 2, respectively.

4.4 Comparison of MSE, PRR, PP and Variance for SRGMs

This study analyses and compares the goodness-of-fit (GOF) and predictive capacity of the proposed SRGM with existing SRGMs. Therefore, we test TE and CP based proposed SRGMs efficiency using two real datasets. Since the proposed SRGM is new for predicting/estimating software reliability, we will evaluate its accuracy with a few well-known SRGMs. However, several GOF criteria have been implemented for demonstrating the model's efficiency and use the NCD method to rank and select the best SRGM. Table 5 shows the comparison of SRGMs by evaluation criteria on real software failure datasets. Both datasets are used to align models and estimate parameters of SRGMs for the suitable power comparison. The results of the estimated MSE, PRR, PP, and variance for DS-1 and DS-2 of all models are summarized in Table 5.

Table 5. Model comparison and model ranking for DS-1 and DS-2.

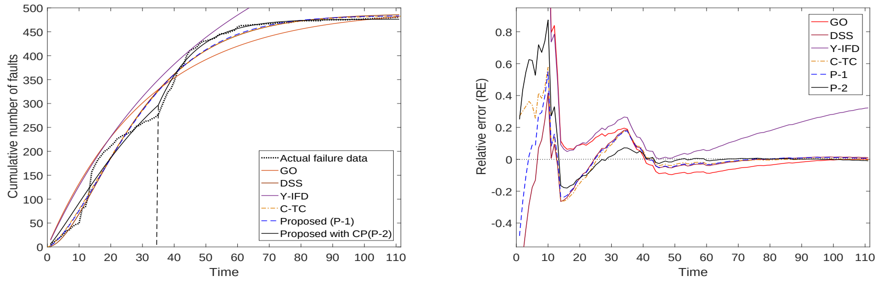
DS	Model	MSE	PRR	PP	Variance	NCD Value	Rank
DS-1	G-O	1006.08	5.11	33.00	31.62	0.204879	7
	DSS	344.20	16.33	2.17	18.18	0.060692	6
	ISS	301.22	1.67	2.49	17.23	0.006791	5
	Yamada-IFD	6000.69	6.46	31.44	132.91	0.865036	9
	Pham-Zhang-IFD	450.15	36.38	4.33	22.40	0.279908	8
	PNZ model	320.40	1.57	2.07	17.82	0.006524	3
	C-TC model	308.299	1.612	2.152	17.56	0.006616	4
	Proposed Model-1	298.659	**1.503**	1.995	17.128	0.006260	2
	Proposed Model-2	**164.714**	4.324	**1.845**	**12.716**	**0.003468**	1
DS-2	G-O	97.89	34.72	5.38	10.63	0.03892	3
	DSS	185.32	6571.6	11.92	15.75	0.360314	8
	ISS	101.31	64.34	6.76	12.18	0.050665	7
	Yamada-IFD	91.50	44.66	5.88	10.21	0.0389526	5
	Pham-Zhang-IFD	187.63	6948.1	12.05	15.73	0.389526	9
	PNZ model	100.29	50.34	6.27	10.05	0.041510	6
	C-TC model	172.60	46.73	8.246	11.62	0.0389314	4
	Proposed Model-1	90.706	4.1126	5.041	8.146	0.029313	2
	Proposed Model-2	**4.2006**	**0.4711**	**0.296**	**1.715**	**0.000467**	1

For DS-1, the proposed model-1 (P-1) has the lowest PRR while the second-lowest MSE, PP and variance value. It is worth noting that the PRR criteria uses the risk of underestimation by assigning a larger penalty to a SRGM that has underestimated the cumulative number of failures. The NCD value is 0.006260, which represents the model ranking as the second. The proposed model-2 (P-2) has the lowest MSE, PP and variance value, while the lowest PRR ranking is fifth. The NCD value is 0.0003468, which represents the model ranking as the first. PRR provides a huge penalty to the SRGM that has slighted the cumulative number of failures. Despite the P-2 does not deliver the best PRR value; however, its notable improvements on MSE, PP, and variation value, we still conclude that the P-2 gives a better prediction concerning the competing SRGMs.

For DS-2, the proposed model-1 has the second-lowest MSE, PRR, PP and variance values. Moreover, it has NCD value as 0.029313, which represents its ranking as a second. Similarly, the proposed model-2 has the lowest MSE, PRR, PP and variance values. Moreover, it has NCD value as 0.000467, which indicates its first rank in all the competing models. The MVF of the proposed and existing models with the observed data are plotted in Figs. 1 (a) and 2 (a). From these figures, it is observed that our proposed models fitting is outperform several competing models. The REs of the SRGMs in terms of the test week shows the comparison with other models for DS-1 and DS-2 is shown in Figs. 1 (b) and 2 (b), respectively. From these figures, we see that P-1 and P-2 reach zero after a certain time.

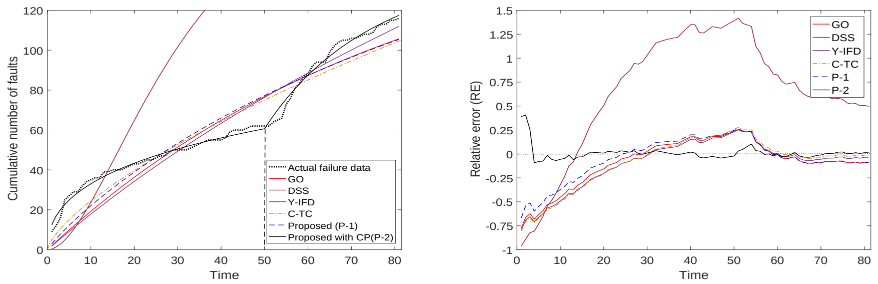
5. CONCLUDING REMARKS

The integration of TE and CP factors during the formulation of SRGMs is more realistic and suitable to describe fault detection. TE consumption can alter overtime during the development process, which is often impacted by certain test constraints. This paper presents a new SRGM based on TE and CP. Two datasets are used to compare the proposed models with several current NHPP models in terms of four evaluation criteria, *i.e.*, MSE, PRR, PP, and variance. Numerical outcomes indicate that better fit and predictive ability can be given by the proposed models. Here, we say better fit based on MSE



(a) (b)

Fig. 1. (a) Cumulative number of faults vs. time and (b) Relative error curve for DS-1.



(a) (b)

Fig. 2. (a) Cumulative number of faults vs. time and (b) Relative error curve for DS-2.

and variation values, while we say predictive ability based on PRR and PP values. An improved NCD approach is adopted to rank and choose the fittest model based on all considered evaluation criteria together. We have tested these models for a few datasets where jump variation of fault detection exists. It may be possible that the proposed models are not outperformed for smooth datasets. More datasets can provide more detailed validation for model outcomes in order to be well documented. The findings showed that the proposed SRGMs would not only getting better-estimated parameters, they also get significantly improved prediction efficiency. All outcomes show that the proposed model can offer dramatically enhanced fit and predictive efficiency. In the future, this work extends with categorical faults based on their severity.

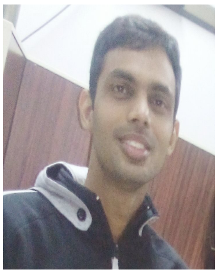
REFERENCES

1. H. Pham, *System Software Reliability*, Springer Science & Business Media, London, 2007.
2. J. Yang, Y. Liu, M. Xie, and M. Zhao, "Modeling and analysis of reliability of multi-release open source software incorporating both fault detection and correction processes," *Journal of Systems and Software*, Vol. 115, 2016, pp. 102-110.

3. S. Chatterjee, A. Shukla, and H. Pham, "Modeling and analysis of software fault detectability and removability with time variant fault exposure ratio, fault removal efficiency, and change point," *Journal of Risk and Reliability*, Vol. 233, 2019, pp. 246-256.
4. M. Zhu and H. Pham, "A two-phase software reliability modeling involving with software fault dependency and imperfect fault removal," *Computer Languages, Systems & Structures*, Vol. 53, 2018, pp. 27-42.
5. K. Y. Song, I. H. Chang, and H. Pham, "A software reliability model with a weibull fault detection rate function subject to operating environments," *Applied Sciences*, Vol. 7, 2017, p. 983.
6. V. Pradhan, J. Dhar, A. Kumar, and A. Bhargava, "An s-shaped fault detection and correction SRGM subject to gamma-distributed random field environment and release time optimization," *Decision Analytics Applications in Industry*, Springer, Singapore, 2020, pp. 285-300.
7. C.-Y. Huang and C.-T. Lin, "Software reliability analysis by considering fault dependency and debugging time lag," *IEEE Transactions on Reliability*, Vol. 55, 2006, pp. 436-450.
8. C.-Y. Huang and W.-C. Huang, "Software reliability analysis and measurement using finite and infinite server queueing models," *IEEE Transactions on Reliability*, Vol. 57, 2008, pp. 192-203.
9. B. Pachauri, J. Dhar, and A. Kumar, "Incorporating inflection S-shaped fault reduction factor to enhance software reliability growth," *Applied Mathematical Modelling*, Vol. 39, 2015, pp. 1463-1469.
10. V. Pradhan, A. Kumar, and J. Dhar, "Modelling software reliability growth through generalized inflection s-shaped fault reduction factor and optimal release time," *Journal of Risk and Reliability*, 2021, p. 1748006X211033713.
11. C.-Y. Huang and M. R. Lyu, "Optimal release time for software systems considering cost, testing-effort, and test efficiency," *IEEE Transactions on Reliability*, Vol. 54, 2005, pp. 583-591.
12. C.-Y. Huang, S.-Y. Kuo, and M. R. Lyu, "An assessment of testing-effort dependent software reliability growth models," *IEEE Transactions on Reliability*, Vol. 56, 2007, pp. 198-211.
13. C.-Y. Huang and S.-Y. Kuo, "Analysis of incorporating logistic testing-effort function into software reliability modeling," *IEEE Transactions on Reliability*, Vol. 51, 2002, pp. 261-270.
14. S.-Y. Kuo, C.-Y. Huang, and M. R. Lyu, "Framework for modeling software reliability, using various testing-efforts and fault-detection rates," *IEEE Transactions on Reliability*, Vol. 50, 2001, pp. 310-320.
15. P. Kapur, D. Goswami, A. Bardhan, and O. Singh, "Flexible software reliability growth model with testing effort dependent learning process," *Applied Mathematical Modelling*, Vol. 32, 2008, pp. 1298-1307.
16. C.-Y. Huang, "Performance analysis of software reliability growth models with testing-effort and change-point," *Journal of Systems and Software*, Vol. 76, 2005, pp. 181-194.

17. C.-T. Lin and C.-Y. Huang, "Enhancing and measuring the predictive capabilities of testing-effort dependent software reliability models," *Journal of Systems and Software*, Vol. 81, 2008, pp. 1025-1038.
18. A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Transactions on Reliability*, Vol. 28, 1979, pp. 206-211.
19. B. Pachauri, A. Kumar, and J. Dhar, "Software reliability growth modeling with dynamic faults and release time optimization using GA and MAUT," *Applied Mathematics and Computation*, Vol. 242, 2014, pp. 500-509.
20. A. K. Shrivastava, V. Kumar, P. Kapur, and O. Singh, "Software release and testing stop time decision with change point," *International Journal of System Assurance Engineering and Management*, Vol. 11, 2020, pp. 196-207.
21. M. Zhao, "Change-point problems in software and hardware reliability," *Communications in Statistics-Theory and Methods*, Vol. 22, 1993, pp. 757-768.
22. K.-C. Chiu, Y.-S. Huang, and T.-Z. Lee, "A study of software reliability growth from the perspective of learning effects," *Reliability Engineering & System Safety*, Vol. 93, 2008, pp. 1410-1421.
23. I. Chang, "An analysis of software reliability with change-point models," NSC 85-2121-M031-003, Taiwan National Science Council, 1997.
24. S. Chatterjee and A. Shukla, "Modeling and analysis of software fault detection and correction process through weibull-type fault reduction factor, change point and imperfect debugging," *Arabian Journal for Science and Engineering*, Vol. 41, 2016, pp. 5009-5025.
25. N. Ahmad, M. G. Khan, and L. S. Rafi, "A study of testing-effort dependent inflection s-shaped software reliability growth models with imperfect debugging," *International Journal of Quality & Reliability Management*, Vol. 27, 2010, pp. 89-110.
26. H.-J. Shyur, "A stochastic software reliability model with imperfect-debugging and change-point," *Journal of Systems and Software*, Vol. 66, 2003, pp. 135-141.
27. S. Chatterjee and A. Shukla, "A unified approach of testing coverage-based software reliability growth modelling with fault detection probability, imperfect debugging, and change point," *Journal of Software: Evolution and Process*, Vol. 31, 2019.
28. C.-Y. Huang and M. R. Lyu, "Estimation and analysis of some generalized multiple change-point software reliability models," *IEEE Transactions on Reliability*, Vol. 60, 2011, pp. 498-514.
29. S. Yamada, M. Ohba, and S. Osaki, "S-shaped software reliability growth models and their applications," *IEEE Transactions on Reliability*, Vol. R-33, 1984, pp. 289-292.
30. M. Ohba, "Inflection S-shaped software reliability growth model," *Stochastic Models in Reliability Theory*, Springer, Berlin,1984, pp. 144-162.
31. S. Yamada, K. Tokuno, and S. Osaki, "Imperfect debugging models with fault introduction rate for software reliability assessment," *International Journal of Systems Science*, Vol. 23, 1992, pp. 2241-2252.
32. H. Pham, L. Nordmann, and Z. Zhang, "A general imperfect-software-debugging model with S-shaped fault-detection rate," *IEEE Transactions on Reliability*, Vol. 48, 1999, pp. 169-175.

33. I. H. Chang, H. Pham, S. W. Lee, and K. Y. Song, "A testing-coverage software reliability model with the uncertainty of operating environments," *International Journal of Systems Science: Operations & Logistics*, Vol. 1, 2014, pp. 220-227.
34. S. Yamada, H. Ohtera, and H. Narihisa, "Software reliability growth models with testing-effort," *IEEE Transactions on Reliability*, Vol. 35, 1986, pp. 19-23.
35. S. Hwang and H. Pham, "Quasi-renewal time-delay fault-removal consideration in software reliability modeling," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 39, 2008, pp. 200-209.
36. Q. Li and H. Pham, "NHPP software reliability model considering the uncertainty of operating environments with imperfect debugging and testing coverage," *Applied Mathematical Modelling*, Vol. 51, 2017, pp. 68-85.



Vishal Pradhan received his M.Sc. degree in 2015 from SAU, Delhi. Currently working as a Ph.D. candidate at ABV-IIITM, Gwalior, India.

Joydip Dhar received his Ph.D. in 1997 from IIT, Kanpur. Actively employed at ABV-IIITM Gwalior, India.

Ajay Kumar received his Ph.D. in 2009 from IIT Roorkee. Currently serves as an Associate Professor at ABV-IIITM, Gwalior, India.