Received May 27, 2019, accepted June 14, 2019, date of publication June 20, 2019, date of current version July 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924084

A Generalized Software Reliability Growth Model With Consideration of the Uncertainty of Operating Environments

QIUYING LI [1,2] AND HOANG PHAM [3], (Fellow, IEEE)
[1] School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China
[2] National Key Laboratory of Science and Technology on Reliability and Environmental Engineering, Beijing 100191, China
[3] Department of Industrial and Systems Engineering, Rutgers University, Piscataway, NJ 08854, USA

Corresponding author: Qiuying Li (li_qiuying@buaa.edu.cn)

This work was supported in part by the National Key Laboratory of Science and Technology on Reliability and Environmental Engineering of China (WDZC2019601A303).

ABSTRACT This paper proposes a generalized model to cover imperfect debugging and the uncertainty of the operating environment and its effect on fault detection rate into software reliability evaluation based on a non-homogeneous Poisson process (NHPP). Many NHPP software reliability growth models (SRGMs) have been developed to estimate the software reliability measures over the past 40 years, but most of these models assume that the operating environment is the same as the testing environment. However, in fact, due to the unpredictability of the uncertain factors in the operating environments for the software, they may considerably influence the software's reliability in an unpredictable way. So when a software system works in a field environment, its reliability is usually different from the original reliability prediction in the testing phase of the software development process, also from all its similar applications in other fields. In this paper, a general model is used to derive models that incorporate the uncertainty of operating environments, which provides the flexibility in considering a different fault detection rate and random environmental factor and so on. Several published models are shown to be covered by this general model and a new model is also developed and examined. The numerical illustrative examples of the proposed model have been validated on two sets of real software failure data in terms of six criteria. The comparison results demonstrate that the new model can fit and predict significantly better than other existing models.

INDEX TERMS Generalized software reliability growth model, non-homogeneous Poisson process, uncertainty, operating environment, imperfect debugging.

I. INTRODUCTION

Software reliability has become one of the most important customer-oriented attributes of software quality [1]. It is of great importance to have effective approaches to develop reliable software along with quantitatively estimating the software reliability [2].

During the past 40 years, many time-dependent NHPP SRGMs have been developed to determine the reliability of software systems [3]–[7]. Different models have been studied upon different assumptions. For instance, some models assume perfect debugging [8], [9], others take into account imperfect debugging [9]–[11], including fault removal efficiency [12], error generation [13]–[15] and learning process [16], [17]. Also some other factors in real developing process are integrated to improve the estimation accuracy of SRGMs, such as testing coverage [18]–[20], testing effort [21]–[25], time-delay fault correction [26]–[29] and fault reduction factor [30]. Most models assume that the operating environment is the same as the testing environment, and the underlying assumption is that the software used in the operating environment has the same failure-occurrence behavior as that used in the software testing environment. So models suitable for the software testing data are also fit for the field data set. But this assumption may not always stand especially for those software will be used in many different field environments after they are released. The in-house testing environment is often a controlled environment with much less variation compared with the field environments, however, the field operation environments for the software differ

The associate editor coordinating the review of this manuscript and approving it for publication was Antonio Pecchia.

considerably from one location to another or one application to another. It has also been noticed that software reliability in the field environment is often different from that in the testing environment, so some certain proportional constant of the environmental factor has been defined to characterize the discrepancy between the testing and field environments from the viewpoint of the severities of usage conditions and several studies on field-oriented software reliability assessment have been conducted [31]–[33]. However, the difficulty in estimating the environmental factor remains an outstanding problem which cannot be solved even in [31]–[33]. On the other hand, the field environment includes much more uncertain factors than the testing environment such as the pattern of the execution load and the operational profile. It is impossible to prepare test cases containing all possible external disturbances. Therefore, the considerations of the randomness of the environmental factor can describe the actual operating environment more faithfully [1]. The uncertainty of the operating environments will greatly influence the software failure and reliability behavior in an unpredictable way.

Teng and Pham firstly proposed a software gain model under the random operating environment with consideration of the effect of the random field environmental factor on the cost model [34]. They also proposed a model that discusses the randomness of the environment and its effects on the fault detection rate under the condition that the uncertainty of operating environmental effect follows the Gamma (or Beta) distribution and fault content function is a linear function of the mean value function [35]. Then Pham presented a new definition of ''systemability'' which is defined as the probability that the system will perform its intended function for a specified mission time subject to the uncertainty of the operating environment [36]. After that, Pham developed a Vtub-shaped software fault detection rate model considering the randomness of the operating environment where the fault-detection rate follows a Vtub-shaped function under the condition that the uncertainty also follows the Gamma distribution and fault content remains a constant [37]. Chang *et al.* introduced the testing coverage into software reliability model considering the uncertainty of operating environment, also assumed that the uncertainty follows the Gamma distribution and the total fault number remains the same [38]. Yamada expanded the meaning of systemability from Pham's definition into the system reliability characteristic considering the uncertainty and the variability of the field operating environment [1]. Inoue *et al.* proposed a bivariate software reliability growth model depending on testing-time and testing-effort with consideration of the uncertainty of a testing-environmental change by assuming that the testing-environmental coefficient follows the Beta distribution, which describes the situation that the testing-environment after change-point is more severe than that before change-point [39]. Recently, we have witnessed several works continuously developed by Song *et al.* in this direction, i.e. they gave a three-parameter fault-detection rate function and assumed that the environmental factor had the exponential distribution [40], a Weibull fault detection rate [41], an S-shaped fault detection rate [42], a fault detection rate function affected by the probability of fault removal on a failure [43], a Weibull failure detection rate function with consideration of testing time with syntax error [44] and an inflection S-shaped fault detection rate [45]. Meanwhile, the last four models all assumed the Gamma distribution environmental factor and a constant expected number of faults existing in the software before testing. Considering the features of isomorphism, Li and Pham developed a model with the total fault number function is a linear function of testing time, along with testing coverage [46].

This paper further proposes a generalized model framework accounting for the uncertainty of operating environments, whose formulation provides the flexibility in modeling the random environmental effects. Under the unified framework, many existing models can be considered as special cases and a new model with Weibull probability density function (pdf) is derived.

The rest of the paper is organized as follows. In Section II, we give an explicit solution to a general class of NHPP SRGMs used to derive the proposed model, and several existing SRGMs are also explained to be special cases of the general model. In Section III, we present the parameter estimation method and criteria for model comparison. In Section IV and V, we compare the performance of this proposed model with several existing NHPP SRGMs based on two sets of software failure data from descriptive and predictive power in terms of six criteria, respectively. Finally, we draw the conclusions in Section VI.

II. SOFTWARE RELIABILITY MODELING

A. A GENERAL NHPP MODEL WITHOUT THE UNCERTAINTY OF OPERATING ENVIRONMENTS

A general class of NHPP SRGMs was proposed by [47] to summarize the existing NHPP models as follows:

$$\frac{dm(t)}{dt} = h(t)\,(a(t) - m(t)) \tag{1}$$

where $m(t)$ means the expected value function of faults detected up to time t, $a(t)$ is the total number of faults in the software at time t, and $h(t)$ represents the fault detection rate function dependent on time.

The general solution of (1) is

$$m(t) = e^{-H(t)}\left[m_0 + \int_{t_0}^{t} a(\tau)h(\tau)e^{H(\tau)}d\tau\right] \tag{2}$$

where

$$H(t) = \int_{t_0}^{t} h(u)\mathrm{d}u \tag{3}$$

and $m(t_0) = m_0$ is the marginal condition of (2) with t_0 representing the starting time of the testing process.

Substituting different $a(t)$ and $h(t)$ into (2), we can get different mean value functions, which represent different assumptions and yield more or less complicated

TABLE 1. Summary of the software reliability models and their mean value functions.

No.	Model name	Model type	Mean value function (m(t))	Comments
1	G-O model[3]	Concave	$m(t)=a(1-\mathrm{e}^{-bt})$	It is also called exponential model.
2	Delayed S-shaped model [48]	S-shaped	$m(t)=a(1-(1+bt)\,\mathrm{e}^{-bt})$	A change of G-O model to make it S-shaped.
3	Inflection S-shaped model [49]	Concave	$m(t)=\dfrac{a(1-\mathrm{e}^{-bt})}{1+\beta\mathrm{e}^{-bt}}$	Special case of G-O model when β equals to 0.
4	HD/G-O model [50]	Concave	$m(t)=\ln[(\mathrm{e}^{a}-c)/(\mathrm{e}^{a\mathrm{e}^{-bt}}-c)]$	Special case of G-O model when c equals to 0.
5	Yamada exponential model [48]	Concave	$m(t)=a(1-\mathrm{e}^{-\gamma\alpha(1-\mathrm{e}^{-\beta t})})$	Taking testing effort into account.
6	Yamada Rayleigh model [48]	S-shaped	$m(t)=a(1-\mathrm{e}^{-\gamma\alpha(1-\mathrm{e}^{-\beta t^{2}/2})})$	Taking testing effort into account.
7	Yamada imperfect 1 model [10]	Concave	$m(t)=\dfrac{ab}{\alpha+b}(\mathrm{e}^{\alpha t}-\mathrm{e}^{-bt})$	Assume that the fault content is an exponential function and the fault detection rate is a constant.
8	Yamada imperfect 2 model [10]	Concave	$m(t)=a(1-\mathrm{e}^{-bt})(1-\dfrac{\alpha}{b})+\alpha at$	Assume that the fault introduction rate is a constant and the fault detection rate is a constant.
9	Pham-Zhang model(1997) [51]	S-shaped and Concave	$m(t)=\dfrac{1}{(1+\beta\mathrm{e}^{-bt})}\left((c+a)(1-\mathrm{e}^{-bt})-\dfrac{ab}{b-\alpha}(\mathrm{e}^{-\alpha t}-\mathrm{e}^{-bt})\right)$	Assume that the fault introduction rate is an exponential function of the testing time and the fault detection rate is non-decreasing.
10	PNZ model(1999) [47]	S-shaped and Concave	$m(t)=\dfrac{a}{(1+\beta\mathrm{e}^{-bt})}\left((1-\mathrm{e}^{-bt})(1-\dfrac{\alpha}{b})+\alpha t\right)$	Assume that the fault introduction rate is a constant and the fault detection rate is non-decreasing.
11	Fault removal model (2003) [12]	S-shaped	$m(t)=\dfrac{a}{p-\beta}\left\{1-\left(\dfrac{(1+\alpha)\mathrm{e}^{-bt}}{1+\alpha\mathrm{e}^{-bt}}\right)^{\frac{c}{b}(p-\beta)}\right\}$	Assume that the fault introduction rate is a constant and the fault detection rate function is non-decreasing.
12	Teng-Pham model (2006) [35]	S-shaped	$m(t)=\dfrac{a}{p-q}\left\{1-\left(\dfrac{\beta}{\beta+(p-q)\ln\left(\dfrac{c+\mathrm{e}^{bt}}{c+1}\right)}\right)^{\alpha}\right\}$	Assume that the fault detection rate function is non-decreasing with an S-shaped curve with 2 parameters.
13	R-M-D model (2014) [15]	Concave	$m(t)=a\alpha\left(1-\mathrm{e}^{-bt}\right)-\dfrac{ab}{b-\beta}\left(\mathrm{e}^{-\beta t}-\mathrm{e}^{-bt}\right)$	Assume that the fault content function is an exponential function of the testing time and constant fault detection rate.
14	Vtub-shaped fault-detection rate model(2014) [37]	S-shaped	$m(t)=N\left(1-\left(\dfrac{\beta}{\beta+a^{t^{b}}-1}\right)^{\alpha}\right)$	Assume constant fault content and the fault detection rate follows a Vtub-shaped function.
15	Chang et al's model (2014) [38]	S-shaped	$m(t)=N\left(1-\left(\dfrac{\beta}{\beta+(at)^{b}}\right)^{\alpha}\right)$	Assume constant fault content and incorporate the testing coverage function into the reliability model.
16	Proposed model	S-shaped	$m(t)=N\left(1-\sum_{n=0}^{\infty}\dfrac{(-1)^{n}}{n!}\Gamma(\dfrac{n+k}{k})\lambda^{n}\left(\dfrac{c}{b}\ln\left(\dfrac{a+\mathrm{e}^{bt}}{1+a}\right)\right)^{n}\right)$	Assume constant fault content and the fault detection rate function is non-decreasing with an S-shaped curve with 3 parameters and η follows Weibull distribution with 2 parameters.

analytic models. Many existing NHPP SRGMs can be taken as a special case of (2).

A constant $a(t)$ means perfect debugging and the total fault number remains the same, and an increasing $a(t)$ means an increasing fault content and implies imperfect debugging.

A constant $h(t)$ means that failure intensity is proportional to the number of remaining faults, and an increasing $h(t)$ means an increasing fault detection rate due to testing learning or an S-shaped $h(t)$ attributed to fluctuations during the testing process [16], [17], or a combination of both mentioned above.

B. A GENERALIZED NHPP MODEL WITH THE UNCERTAINTY OF OPERATING ENVIRONMENTS

Based on the following basic assumptions, an NHPP SRGM considering the uncertainty of operating environments is proposed:

(1) Software faults' occurrence and removal follow an NHPP.

(2) The software failure rate at any time is proportional to the number of faults remaining in the software at that time.

(3) When a software failure is detected, the fault causing the failure will be removed immediately.

TABLE 2. Failure data of DS-1.

Testing time (months)	Faults	Cumulative detected faults	Testing time (months)	Faults	Cumulative detected faults	Testing time (months)	Faults	Cumulative detected faults	Testing time (months)	Faults	Cumulative detected faults
1	1	1	16	3	30	31	0	53	46	0	84
2	6	7	17	15	45	32	1	54	47	0	84
3	0	7	18	2	47	33	2	56	48	0	84
4	2	9	19	2	49	34	2	58	49	1	85
5	0	9	20	2	51	35	1	59	50	1	86
6	0	9	21	2	53	36	1	60	51	3	89
7	3	12	22	0	53	37	3	63	52	1	90
8	6	18	23	0	53	38	7	70	53	0	90
9	0	18	24	0	53	39	1	71	54	2	92
10	0	18	25	0	53	40	0	71	55	16	108
11	0	18	26	0	53	41	1	72	56	12	120
12	3	21	27	0	53	42	2	74	57	8	128
13	1	22	28	0	53	43	0	74	58	1	129
14	0	22	29	0	53	44	6	80	59	10	139
15	5	27	30	0	53	45	4	84	60	7	146

(4) The environment affects the fault detection rate, $h(t)$, by a random variable η.

The assumption (4) is suggested by Teng and Pham [34], and they have suggested a model that captures the uncertainty of field environments by multiplying the fault detection rate with a factor η, where η is a random variable. Then the model can be given as follows:

$$\frac{dm(t)}{dt} = \eta h(t)\left(a(t) - m(t)\right) \tag{4}$$

Assume η's pdf is $g(\eta)$ and η is a time-independent variable and unit-free.

From (4), we can get the generalized solution given in term of η as follows:

$$m_\eta(t) = e^{-\eta H(t)}\left[m_0 + \int_{t_0}^{t} \eta a(\tau)h(\tau)e^{\eta H(\tau)}d\tau\right] \tag{5}$$

From (5), we can obtain the mean value function as follows:

$$\begin{aligned} m(t) &= E(m_\eta(t)) = \int_0^{+\infty} m_x(t)g(x)dx \\ &= \int_0^{+\infty} e^{-xH(t)}(m_0 + \int_{t_0}^{t} xa(\tau)h(\tau)e^{xH(\tau)}d\tau)g(x)dx \end{aligned} \tag{6}$$

Solving (6) using the initial condition that at $t_0 = 0$, $m(t_0) = m(0) = 0$, and the closed-form solution $m(t)$ is given as follows:

$$\begin{aligned} m(t) &= \int_0^{+\infty} e^{-xH(t)}(\int_0^{t} xa(\tau)h(\tau)e^{xH(\tau)}d\tau)g(x)dx \\ &= a(t) - a(0)F^*(H(t)) - \sum_{n=0}^{+\infty}\frac{(-1)^n}{n!}F^{(n)*}(H(t)) \\ &\quad \times \left[a(t)H^n(t) - n\int_0^{t} a(\tau)h(\tau)H^{n-1}(\tau)d\tau\right] \end{aligned} \tag{7}$$

where

$$H(t) = \int_0^t h(u)\mathrm{d}u \tag{8}$$

$F^*(H(t))$ means the Laplace transform of the pdf $g(x)$ and $F^{(n)*}(H(t))$ represents the n^{th} order differential of the Laplace transform of $g(x)$ (The detailed demonstration of (7) can be seen in Appendix A).

Substituting different type of pdf $g(\eta)$, the total fault number function $a(t)$ and the fault detection rate function $h(t)$, we can get different mean value functions correspondingly. It should be noted that any nonnegative distribution could be used to model the random environmental factor η. Here most published models assume that the environmental factor η follows the Gamma distribution because of its great flexibility, which can include all conditions between operational environment and testing environment where the usage condition is more favorable to fault detection than the testing environment, i.e. ($\eta > 1$), or equivalent to the testing environment, i.e. ($\eta = 1$), or less favorable to fault detection than the testing environment, i.e. ($0 < \eta < 1$). But note that here $g(\eta)$, $a(t)$, $h(t)$ are all arbitrary functions without any restricted limitations, that is, not limited to the condition that η has the Gamma distribution and fault content function is a linear function of mean value function.

When $a(t) = a + qm(t)$, (7) can be simplified as the following equation:

$$m(t) = \frac{a}{1-q}\left(1 - F^*((1-q)H(t))\right) \tag{9}$$

where $a = a(0)$, refers to the total number of faults that exist in the software before testing; $a(t)$ refers to the total number of faults in the software at time t, which is proportional to the mean value function of $m(t)$.

Now we are going to discuss some existing models from the general viewpoint according to (7) and (9), then we will develop a new model along the similar line.

TABLE 3. Comparison of SRGMs' descriptive power for DS-1.

Model No.	Model Name	Model Parameter Estimation Results	MSE(fit)	R^2 (fit)	Adjusted R^2 (fit)	PP(fit)	AIC(fit)
1	G-O model	$\hat{a}$ = 1.414e+04 , $\hat{b}$ = 1.364e-04	86.8103	0.9293	0.9281	2.1979	372.5853
2	Delayed S-shaped model	$\hat{a}$ = 153.2 , $\hat{b}$ = 0.04338	135.6552	0.8895	0.8876	6.3987	417.9063
3	Inflection S-shaped model	$\hat{a}$ = 1870 , $\hat{b}$ = 0.001127 , $\hat{\beta}$ = 0.07244	89.8070	0.9281	0.9256	2.2768	376.5951
4	HD/G-O model	$\hat{a}$ = 107.5 , $\hat{b}$ = 0.03036 , $\hat{c}$ = 310	230.8772	0.8152	0.8087	10.7374	491.6726
5	Yamada exponential model	$\hat{a}$ = 253.7 , $\hat{\beta}$ = 0.008351 $\hat{\gamma}$ = 1.037 , $\hat{\alpha}$ = 1.24	150.5000	0.8817	0.8754	6.0308	428.6627
6	Yamada Rayleigh model	$\hat{a}$ = 147.8 , $\hat{\beta}$ = 0.001453 $\hat{\gamma}$ = 1.157 , $\hat{\alpha}$ = 0.9904	198.2143	0.8442	0.8359	6.3032	528.9353
7	Yamada imperfect 1 model	$\hat{a}$ = 28.99 , $\hat{b}$ = 0.1119 $\hat{\alpha}$ = 0.0285	58.0526	0.9535	0.9519	6.8115	342.0316
8	Yamada imperfect 2 model	$\hat{a}$ = 2532 , $\hat{b}$ = 0.0006825 $\hat{\alpha}$ = 0.005719	84.3158	0.9325	0.9302	1.9784	365.6172
9	Pham-Zhang model(1997)	$\hat{a}$ = 825 , $\hat{b}$ = 0.008143 , $\hat{c}$ = 724.1 $\hat{\alpha}$ = 1.035e+04 , $\hat{\beta}$ = 6.489	87.1091	0.9328	0.9279	1.9827	367.4629
10	PNZ model(1999)	$\hat{a}$ = 666.4 , $\hat{b}$ = 0.003515 $\hat{\alpha}$ = 0.006929 , $\hat{\beta}$ = 0.3607	85.9286	0.9325	0.9288	1.9766	367.5774
11	Fault removal model (2003)	$\hat{a}$ = 2175 , $\hat{\alpha}$ = 1407 , $\hat{b}$ = 4.067 $\hat{p}$ = 0.9041 , $\hat{c}$ = 0.0009375 , $\hat{\beta}$ = 0.1246	97.3519	0.9262	0.9194	3.6433	396.2171
12	Teng-Pham model (2006)	$\hat{a}$ = 214.6 , $\hat{\alpha}$ = 0.177 , $\hat{b}$ = 0.175 $\hat{p}$ = 0.6887 , $\hat{c}$ = 8.56e-08 $\hat{\beta}$ = 3.973 , $\hat{q}$ = 0.7946	89.0377	0.9338	0.9263	1.9941	368.6747
13	R-M-D model (2014)	$\hat{a}$ = 118 , $\hat{\alpha}$ = 6.131 $\hat{b}$ = 0.00284 , $\hat{\beta}$ = 1.038	95.1786	0.9252	0.9212	2.1217	384.1152
14	Vtub-shaped fault-detection rate model(2014)	$\hat{N}$ = 2.107e+04 , $\hat{a}$ = 1.235 , $\hat{\alpha}$ = 0.9736 $\hat{b}$ = 0.5568 , $\hat{\beta}$ = 1151	76.6182	0.9408	0.9365	5.9487	408.2347
15	Chang et al's model (2014)	$\hat{N}$ = 3390 , $\hat{a}$ = 0.4908 , $\hat{\alpha}$ = 0.02086 $\hat{b}$ = 1.499 , $\hat{\beta}$ = 41.43	117.7455	0.9091	0.9025	2.6399	7.1769e+03
16	Proposed model	$\hat{N}$ = 72.65 , $\hat{a}$ = 3.878 , $\hat{b}$ = 0.334 $\hat{c}$ = 0.1132 , $\hat{k}$ = 237.6 , $\hat{\lambda}$ = 0.511	**27.8704**	**0.9789**	**0.9769**	**1.6281**	**319.9958**

Notes: The bold number means the result of the best SRGM in this column.

Model-1: Let $a(t) = a$, $h(t) = b\ln(a)t^{b-1}a^{t^b}$, $g(x) = \frac{\beta^\alpha x^{\alpha-1}e^{-\beta x}}{\Gamma(\alpha)}$ (for $\alpha, \beta > 0; x \geq 0$),

$$g(x) = \frac{\beta^\alpha x^{\alpha-1}e^{-\beta x}}{\Gamma(\alpha)} \quad (\text{for } \alpha, \beta > 0; x \geq 0)$$

where the fault content $a(t)$ remains the same as the testing progresses, which implies perfect debugging, the fault detection rate is Vtub-shaped, and η is assumed to have the Gamma pdf, then substitute all above values into (9), we can get

$$m(t) = a\left[1 - \left(\frac{\beta}{\beta + a^{t^b} - 1}\right)^\alpha\right] \tag{10}$$

which is the same as the model given by Pham [37].

Model-2: Let $a(t) = a$, $h(t) = \frac{c'(t)}{1-c(t)}$, $c(t) = 1 - e^{-(at)^b}$, and η follows the Gamma distribution, then substitute all above values into (9), we can get

$$m(t) = a\left[1 - \left(\frac{\beta}{\beta + (at)^b}\right)^\alpha\right] \tag{11}$$

which is the same as the model given by [38].

Model-3: Let $a(t) = a + qm(t)$, $h(t) = \frac{b}{1+ce^{-bt}}$, and η is also assumed to have the Gamma pdf, then according to (9), we can get

$$m(t) = \frac{a}{1-q}\left\{1 - \left(\frac{\beta}{\beta + (1-q)\ln\left(\frac{c+e^{bt}}{c+1}\right)}\right)^\alpha\right\} \tag{12}$$

which is the same as the model given by [35] when $p = 1$ (where p represents the fault removal efficiency $(0 < p \leq 1)$) and $T = 0$ (where T represents the time to stop testing and release the software for field operations).

Model-4: Let $a(t) = a$, $h(t) = \frac{b}{1+ce^{-bt}}$, and η is also assumed to follow Gamma distribution, then according to (9),

TABLE 4. Failure data of DS-2.

Day No.	Faults	Cumulative faults	Day No.	Faults	Cumulative faults	Day No.	Faults	Cumulativ e faults	Day No.	Faults	Cumulative faults
1	2	2	20	6	124	39	5	204	58	3	303
2	0	2	21	2	126	40	1	205	59	4	307
3	2	4	22	3	129	41	4	209	60	8	315
4	3	7	23	2	131	42	6	217	61	3	318
5	3	10	24	3	134	43	3	220	62	4	322
6	6	16	25	8	142	44	2	222	63	5	327
7	8	24	26	6	148	45	6	228	64	6	333
8	8	32	27	7	155	46	13	241	65	0	333
9	12	44	28	8	163	47	9	250	66	4	337
10	10	54	29	2	165	48	6	256	67	5	342
11	6	60	30	3	168	49	7	263	68	4	346
12	5	65	31	4	172	50	3	266	69	5	351
13	4	69	32	3	175	51	3	269	70	5	356
14	6	75	33	3	178	52	4	273	71	5	361
15	10	85	34	4	182	53	5	278	72	3	364
16	6	91	35	4	186	54	6	284	73	3	367
17	7	98	36	5	191	55	6	290			
18	10	108	37	4	195	56	5	295			
19	10	118	38	4	199	57	5	300			

we can get

$$m(t) = a\left\{1-\left(\frac{\beta}{\beta+\ln\left(\frac{c+e^{bt}}{c+1}\right)}\right)^{\alpha}\right\} \tag{13}$$

which is the same as the model given by [45].

Model-5: Let $a(t) = a$, $h(t) = \frac{b}{1+ce^{-kt}}$, and η is assumed to follow the exponential distribution with parameter β, i.e. $g(x) = \beta e^{-\beta x}$ (for $\beta > 0; x \geq 0$), then according to (9), we can get

$$m(t) = a\left\{1-\left(\frac{\beta}{\beta-\frac{b}{k}\ln\left(\frac{(1+c)e^{-kt}}{1+ce^{-kt}}\right)}\right)\right\} \tag{14}$$

which is the same as the model given by [40].

Model-6: Now we will derive a new model from the general class of models denoted as (7).

Assume that the fault content $a(t)$ remains the same as the testing progresses, which means

$$a(t) = N \tag{15}$$

where N denotes the initial fault number presented in the software system before testing starts. That means each time when a failure occurs, the faults causing the failure are removed instantaneously and no new faults are introduced (i.e. perfect debugging).

Let $h(t) = \frac{c}{1+ae^{-bt}}$, where fault detection rate $h(t)$ is a non-decreasing function with an inflection S-shaped curve with 3 parameters, When t tends to infinite, $h(t)$ tends to get its maximum value, then

$$H(t) = \int_0^t \frac{c}{1+ae^{-bu}}du = \frac{c}{b}\ln\left(\frac{a+e^{bt}}{1+a}\right) \tag{16}$$

Let η take the Weibull pdf with 2 parameters, i.e. $g(x) = \frac{k}{\lambda}(\frac{x}{\lambda})^{k-1}e^{-(\frac{x}{\lambda})^k}$ (for $k, \lambda > 0; x \geq 0$).

Substituting all above values into (7), we can obtain the mean value function for the proposed model as follows:

$$m(t)=N\left(1-\sum_{n=0}^{+\infty}\frac{(-1)^n}{n!}\Gamma\left(\frac{n+k}{k}\right)\lambda^n\left(\frac{c}{b}\ln\left(\frac{a+e^{bt}}{1+a}\right)\right)^n\right) \tag{17}$$

(The detailed demonstration of (17) can be seen in Appendix B).

The software reliability function based on the NHPP is shown as follows:

$$R(x/t) = e^{-[m(t+x)-m(t)]} \tag{18}$$

where $R(x/t)$ refers to the software reliability function by time t for a mission time x and $m(t)$ is given by (17).

Table I gives a summary of the proposed model and 15 existing ones. In the next sections, we examine the descriptive and predictive properties of the proposed model by comparing it with these existing models.

III. PARAMETER ESTIMATION AND CRITERIA FOR MODEL COMPARISONS

A. PARAMETER ESTIMATION METHOD

Theoretically, once the analytical expression for $m(t)$ is derived, the parameters in $m(t)$ can be estimated by using the

TABLE 5. Comparison of SRGMs' descriptive power for DS-2.

Model No.	Model Name	Model Parameter Estimation Results	MSE(fit)	R^2 (fit)	Adjusted R^2 (fit)	PP(fit)	AIC(fit)
1	G-O model	$\hat{a}=1663$, $\hat{b}=0.003445$	47.8028	0.996	0.996	46.7777	345.2340
2	Delayed S-shaped model	$\hat{a}=409.8$, $\hat{b}=0.04608$	204.0845	0.9831	0.9829	2.1443	357.0219
3	Inflection S-shaped model	$\hat{a}=1212$, $\hat{b}=0.006456$, $\hat{\beta}=0.3726$	48.4714	0.9961	0.9959	46.2422	634.2445
4	HD/G-O model	$\hat{a}=709$, $\hat{b}=0.009347$, $\hat{c}=10.17$	127.7714	0.9896	0.9893	67.3251	353.1304
5	Yamada exponential model	$\hat{a}=3273$, $\hat{\beta}=0.00373$, $\hat{\gamma}=0.748$, $\hat{\alpha}=0.6607$	58.1449	0.9953	0.9951	53.2839	349.7289
6	Yamada Rayleigh model	$\hat{a}=414.5$, $\hat{\beta}=0.0007953$, $\hat{\gamma}=1.248$, $\hat{\alpha}=1.401$	424.3478	0.9659	0.9644	3.6692	417.4604
7	Yamada imperfect 1 model	$\hat{a}=983.4$, $\hat{b}=0.005853$, $\hat{\alpha}=0.002001$	48.5143	0.996	0.9959	47.2951	347.3469
8	Yamada imperfect 2 model	$\hat{a}=668.1$, $\hat{b}=0.00863$, $\hat{\alpha}=0.004599$	48.5429	0.996	0.9959	47.4813	347.3902
9	Pham-Zhang model (1997)	$\hat{a}=333$, $\hat{b}=0.09861$, $\hat{c}=0.5692$, $\hat{\alpha}=39.96$, $\hat{\beta}=21.22$	729.7059	0.9423	0.9389	4.6897	496.0981
10	PNZ	$\hat{a}=100$, $\hat{b}=0.08265$, $\hat{\alpha}=0.04423$ $\hat{\beta}=0.7343$	44.2319	0.9964	0.9963	31.9915	344.9557
11	Fault removal model (2003)	$\hat{\beta}=0.04763$, $\hat{a}=782.5$, $\hat{\alpha}=406.9$, $\hat{b}=3.465$, $\hat{c}=0.008084$, $\hat{p}=0.7922$	39.6269	0.9969	0.9967	4.7717	348.1968
12	Teng-Pham model (2006)	$\hat{a}=845.2$, $\hat{\alpha}=0.2277$, $\hat{b}=0.3473$, $\hat{\beta}=10.49$, $\hat{c}=0.8023$, $\hat{p}=0.6046$, $\hat{q}=0.4331$	43.4545	0.9967	0.9964	19.9951	346.4750
13	R-M-D model (2014)	$\hat{a}=1086$, $\hat{\alpha}=1.0$, $\hat{b}=0.6339$, $\hat{\beta}=0.005773$	39.3768	0.9968	0.9967	9.2158	3215
14	Vtub-shaped fault-detection rate model (2014)	$\hat{N}=632.1$, $\hat{a}=1.054$, $\hat{\alpha}=0.3546$, $\hat{b}=0.9456$, $\hat{\beta}=1.987$	60.8676	0.9952	0.9949	22.1435	372.8426
15	Chang et al's model (2014)	$\hat{N}=2607$, $\hat{a}=2.564$, $\hat{\alpha}=0.251$ $\hat{b}=1.148$, $\hat{\beta}=492.8$	47.5882	0.9962	0.996	23.6474	5549.3
16	Proposed model	$\hat{N}=1108$, $\hat{a}=38.83$, $\hat{b}=1.289$ $\hat{c}=0.03287$, $\hat{k}=0.8445$, $\hat{\lambda}=0.1884$	**33.7612**	**0.9974**	**0.9972**	**1.5925**	**340.3372**

Notes: The bold number means the result of the best SRGM in this column.

maximum likelihood estimation (MLE) method or the least square estimation (LSE) method. MLE is one of the most useful techniques for deriving estimators because comparing to other estimation methods the maximum likelihood estimates are consistent and asymptotically normally distributed as the sample size increases [52]. However, sometimes the estimations may not be obtained by MLE especially under some conditions where $m(t)$ is too complex, and we need to turn to LSE. So here we use LSE method to estimate the models' parameters.

Let all the failure data are expressed in the form of pairs (t_i, y_i) $(i = 1, 2, \ldots, n; 0 < t_1 < t_2 < \cdots < t_n)$, where y_i is the cumulative number of faults detected in time $(0, t_i]$. The sum of the squared distance is given as follows:

$$Q = \sum_{i=1}^{n} (y_i - m(t_i))^2 \tag{19}$$

By taking derivatives of (19) with respect to each parameter in (17), and setting the results equal to zero, we can obtain seven equations for the proposed model as follows:

$$\frac{\partial Q}{\partial N} = \frac{\partial Q}{\partial a} = \frac{\partial Q}{\partial b} = \frac{\partial Q}{\partial c} = \frac{\partial Q}{\partial k} = \frac{\partial Q}{\partial \lambda} = 0 \tag{20}$$

After solving the above equations simultaneously, we can obtain the least square estimates of all parameters for the proposed model.

As noted, solutions of (20) are extremely difficult and require either graphical or numerical methods. Under the help of developed MATLAB programs based on LSE method, the calculation of parameters is not a critical problem though adding additional parameters to make the software reliability model more complex and the work of parameter estimation more difficult.

TABLE 6. Comparison of G-O, Vtub model, Teng & Pham model and the proposed model using DS-1.

Testing time (months)	Cumulative faults	Predicted total defects by G-O	Predicted total defects by Vtub	Predicted total defects by Teng & Pham	Predicted total defects by the proposed model
1	1	-	-	-	-
2	7	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
50	86	-	-	-	-
51	89	86.0803	84.1576	84.9930	93.9811
52	90	87.6770	85.4286	85.3875	96.2591
53	90	89.1368	86.8783	89.3457	95.2420
54	92	90.4348	88.5492	89.6783	97.2751
55	108	91.8296	89.5973	91.1256	99.2781
56	120	95.0759	92.5570	94.9270	106.1296
57	128	99.5947	99.7037	99.4934	114.4692
58	129	104.9037	101.6265	104.6687	123.5078
59	139	109.9030	107.2100	109.5992	131.8813
60	146	113.6113	109.9667	116.0373	140.2609
Predicted MSE		522.8750	1.0034e+03	1.3745e+03	**171.1531**
Predicted AIC		283.5031	289.6609	294.0642	**280.1920**
Predicted PP		0.2449	0.2966	0.2454	**0.0517**

Notes: The bold number means the result of the best SRGM in this column.

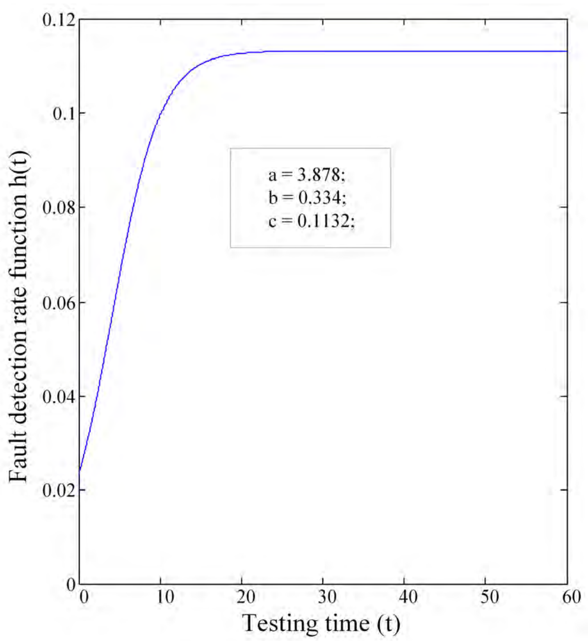

FIGURE 1. Fault detection rate function $h(t)$ for DS-1.

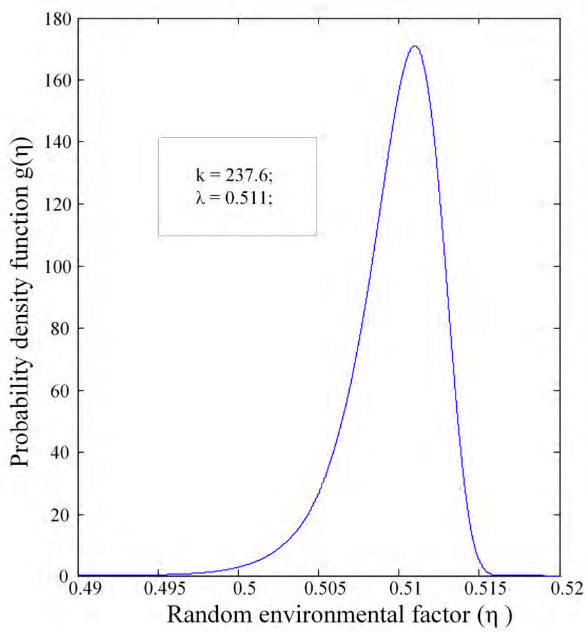

FIGURE 2. Probability density function $g(\eta)$ for DS-1.

B. COMPARISON CRITERIA

Firstly, we use five commonly used goodness-of-fit criteria to examine the descriptive power for all models. The first criterion is the mean value of squared error (Mean Square-Error, MSE), which is defined as follows [2], [37], [38], [53]:

$$MSE = \frac{1}{n-N}\sum_{i=1}^{n}(y_i - \hat{m}(t_i))^2 \tag{21}$$

where n is the number of observations, y_i is the total number of faults detected up to time t_i in terms of the testing data, $\hat{m}(t_i)$ is the estimated value of cumulative fault number up to time t_i according to the fitted mean value function, $i = 1, 2, \ldots, n$.N represents the number of parameters used in the model.

When comparing the performance of models with different numbers of parameters, it is always considered unfair

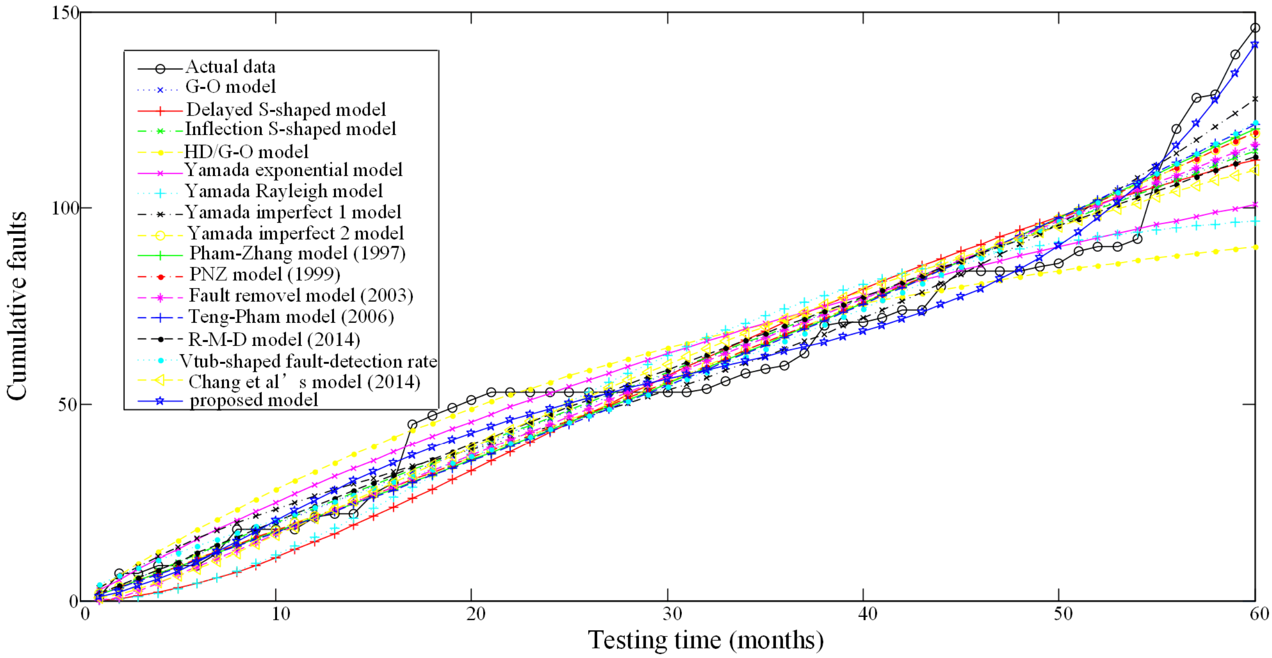

FIGURE 3. **The fitting results of comparison SRGMs compared with actual data for DS-1.**

to simply compare the performance of models with more parameters with others with fewer parameters without giving any penalty to those models with more parameters.

It should be noted that MSE considers the penalty term with respect to the degrees of freedom when there are many parameters and assigns a larger penalty to a model with more parameters. Thus, smaller value of MSE indicates better goodness-of-fit.

The second criterion which is used to examine the fitting power of SRGMs is correlation index of the regression curve equation (R^2), which is expressed as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \hat{m}(t_i)\right)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{22}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Therefore, the larger R^2, the better is the model's performance.

The third criterion which is used to evaluate the fitting performance of SRGMs is adjusted R^2 (Adjusted R^2), which can be expressed as follows:

$$AdjustedR^2 = 1 - \frac{(1-R)(n-1)}{n-P-1} \tag{23}$$

where R denotes the value of R^2 and P represents the number of predictors in the fitted model. Therefore, the larger Adjusted R^2, the better is the model's goodness-of-fit.

The fourth criterion which is used to evaluate the performance of SRGMs is the predictive power (PP), which measures the distance of the estimation given by the model from the actual data against the actual data and defined as follows:

$$PP = \sum_{i=1}^{n} \left(\frac{\hat{m}(t_i) - y_i}{y_i}\right)^2 \tag{24}$$

Therefore, the less the value of PP, the better the model fits.

The fifth criterion is AIC, which measures the ability of a model to maximize the likelihood function that is directly related to the degrees of freedom during fitting and defined as follows:

$$AIC = -2\log L + 2N \tag{25}$$

where L is the maximum value of likelihood function, N represents the number of parameters used in the model.

The AIC criterion also takes the degrees of freedom into consideration by assigning a model with more parameters a larger penalty. So the AIC criterion is regarded as a fair criterion when comparing models with more parameters or few parameters. Thus, the lower value of AIC indicates better goodness-of-fit.

Then we use MSE criterion to examine the prediction performance of SRGMs. But here MSE criterion for prediction has the following definition:

$$MSE_{prediction} = \frac{1}{n-m+1-N}\sum_{i=m}^{n} (y_i - \hat{m}(t_i))^2 \tag{26}$$

Assume that by the end of testing time t_n, totally y_n faults have been detected. Firstly we use the data points up to time $t_{m-1}(t_{m-1} < t_n)$ to estimate the parameters of $m(t)$, then substituting the estimated parameters into the mean value

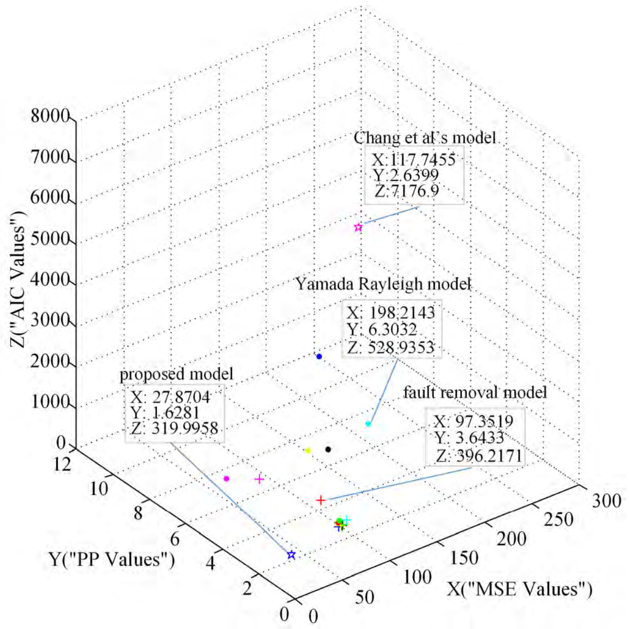

FIGURE 4. A three-dimension plot (X,Y,Z) represents (MSE, PP, AIC) values for DS-1.

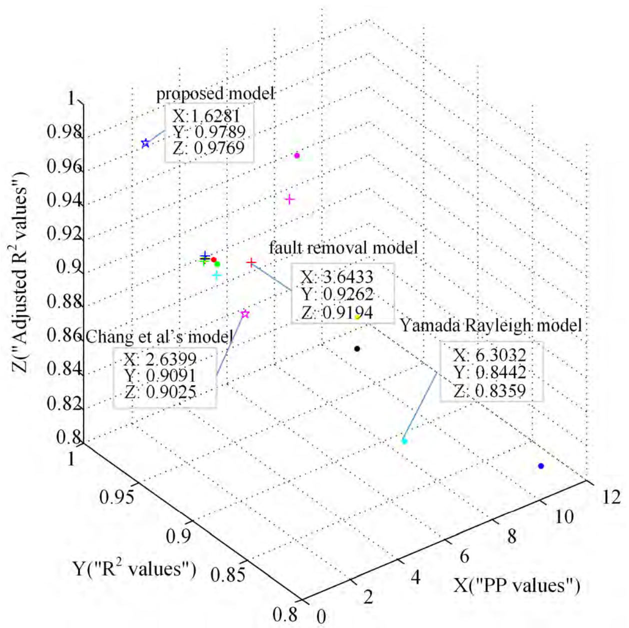

FIGURE 5. A three-dimension plot (X,Y,Z) represents (PP, R2, Adjusted R2) values for DS-1.

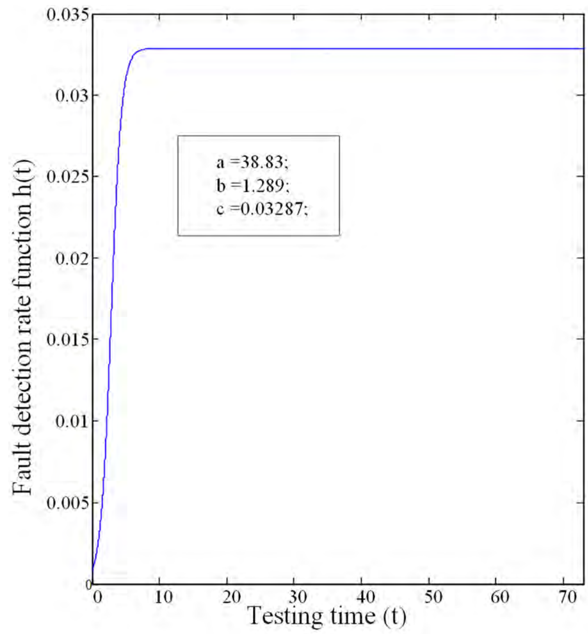

FIGURE 6. Fault detection rate function $h(t)$ for DS-2.

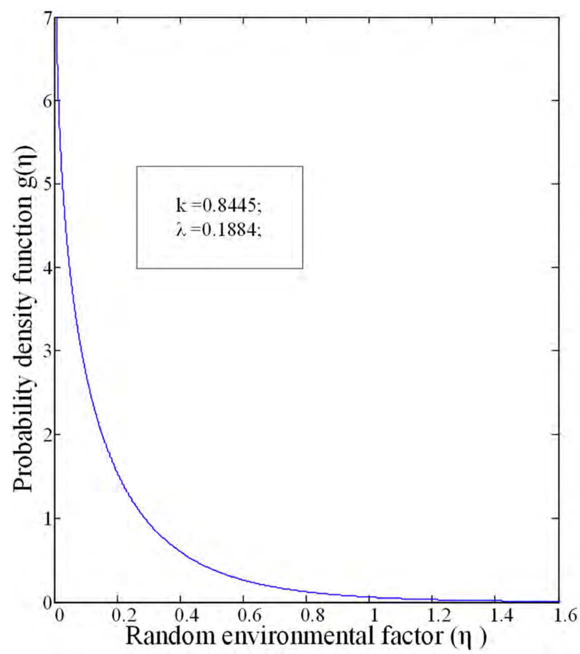

FIGURE 7. Probability density function $g(\eta)$ for DS-2.

function yields the prediction value of the cumulative fault number $\hat{m}(t_i)$ $(i = m+1, m+2, \ldots, n)$. N also refers to the number of parameters used in the model. Thus, the lower value of $MSE_{prediction}$ indicates better predictive power.

IV. NUMERICAL EXAMPLES OF COMPARISON OF MODELS' FITTING POWER

To validate the robustness of the proposed model, we evaluate the fitting power of the proposed model and several existing NHPP models on two different data sets, one is from historical fault data set, the other is from real-world reliability data set collected from real operation.

A. DATA FROM WEBERP SYSTEM

The first historical fault data set is from a web-based integrated accounting ERP system (WebERP) from August 2003 to July 2008 (Data Set 1, DS-1) [54]. This data set is one widely used data set in lots of papers, such as [55], [56]. The details are recorded in Table II and the time unit is month. There are totally 146 faults observed within 60 months.

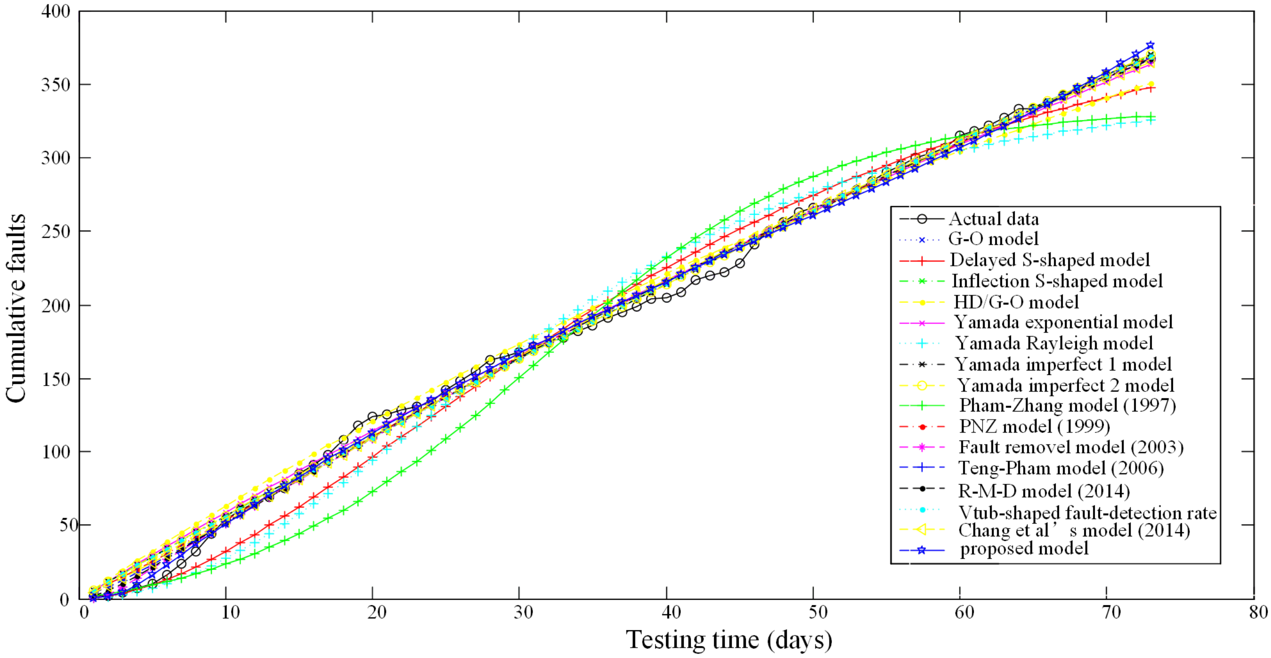

FIGURE 8. The fitting results of comparison SRGMs compared with actual data for DS-2.

For the fitting power comparison, all data points are used to fit the models and estimate the models' parameters. Column III in Table III lists the parameters estimated by LSE method, and MSE values, R^2 values, Adjusted R^2 values, PP values and AIC values based on the parameters estimated by LSE are shown from Column IV to Column VIII in Table III, respectively.

From Table III, it can be seen that in the context of LSE method the proposed model gives the following value:

- AIC = 319.9958, which is the smallest value among all models' values. Others' AIC values are all bigger than it, e.g. Yamada imperfect 1 model's AIC is 342.0316 and 1.07 times larger than the value of the proposed model, even Chang et al's model's AIC value is 7.1769e+03 and 22.43 times larger than the value of the proposed model.
- MSE = 27.8704, which is the smallest value compared to all models' MSE values and also significantly smaller than the values of other models. Others' can be 2.08 times (Yamada imperfect 1 model's 58.0526), even 8.28 times (HD/G-O model's 230.8772) larger than the value of the proposed model.
- R^2 = 0.9789, which is the largest value among all models'.
- Adjusted R^2 = 0.9769, which is the largest value among all models'.
- PP = 1.6281, which is the lowest value among all models'.

This is encouraging that it indicates that though the proposed model is more complicated with 6 parameters, but it has all the best values in terms of the five criteria in the context of LSE method. The excellent performance to describe the debugging process of the proposed model can compensate its complexity.

The fault-detection rate function $h(t)$ and the pdf $g(\eta)$ for DS-1 of the proposed model are graphically illustrated in Fig.1 and Fig.2. The fitting comparison of all models for DS-1 is shown in Fig.3. The coordinates X, Y and Z in Fig.4 represent the MSE, PP and AIC values of all models for DS-1, respectively. The coordinates X, Y and Z in Fig.5 represent the PP, R^2 and Adjusted R^2 values of all models for DS-1, respectively.

B. DATA FROM A MILITARY DIGITAL CONTROL SYSTEM

To validate the effectiveness of the proposed model in the actual usage, we use one real-world reliability data set (Data Set 2, DS-2) reported by [57]. This data set was collected from a military control system by system testing and testing on board. Table IV presents the details of the failure data and the time unit is day. There are totally 367 faults detected within 73 days. For the descriptive power comparison, all data points are used to fit the models and estimate the parameters. Here the model parameters are estimated by LSE method and shown in the third column of Table V. The MSE values, R^2 values, Adjusted R^2 values, PP values and AIC values are listed from Column IV to Column VIII in Table V, respectively. From Table V, it can be seen that for the proposed model:

- MSE = 33.7612 is the smallest value, which is significantly smaller than other models' values from 39.3768 to 729.7059, which means other models' values are from 1.17 to 22.61 times bigger than the value of the proposed model.

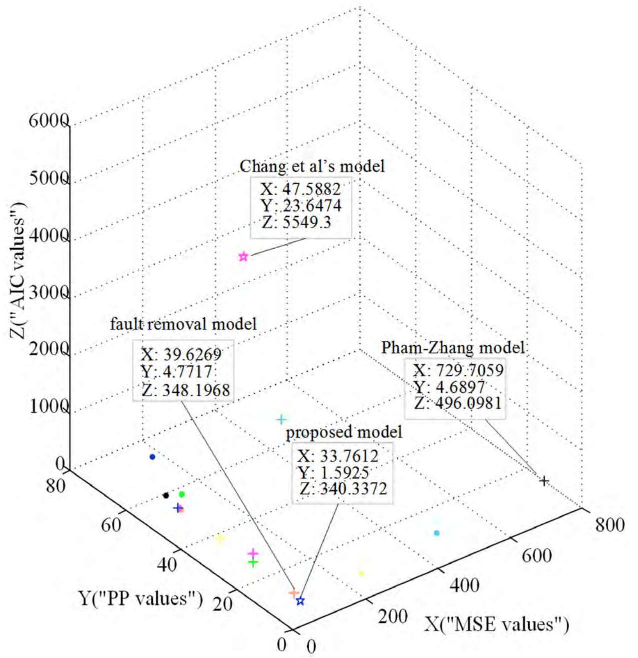

FIGURE 9. A three-dimension plot (X,Y,Z) represents (MSE, PP, AIC) values for DS-2.

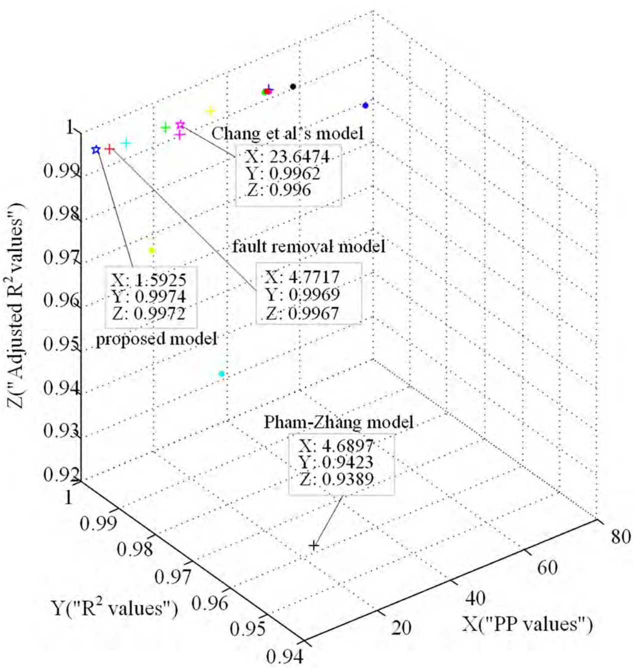

FIGURE 10. A three-dimension plot (X,Y,Z) represents (PP, R^2, Adjusted R^2) values for DS-2.

- Similarly, $R^2 = 0.9974$ is the largest value.
- Adjusted $R^2 = 0.9972$ is the largest value.
- PP $= 1.5925$ is the smallest value.
- AIC $= 340.3372$, which is the smallest value among all models' values.

Thus the proposed model has the best fitting results with all best criteria values and is the best model among all models.

The fault-detection rate function $h(t)$ and the pdf $g(\eta)$ for DS-2 of the proposed model is graphically illustrated in Fig.6 and Fig.7, respectively. The fitting comparison of all models for DS-2 is shown in Fig.8. The coordinates X, Y and Z in Fig.9 illustrate the values of MSE, PP and AIC respectively and the coordinates X, Y and Z in Fig.10 represent the PP, R^2 and Adjusted R^2 values of all models for DS-2. From all figures, we can also observe that the new model shows the best fitting power on the real data set than all other models.

V. NUMERICAL EXAMPLES OF COMPARISON OF MODELS' PREDICTING POWER

In order to validate the performance of the proposed model's predictive power and due to space limited, we only take DS-1 for example. We predict the cumulative of faults from Month 51 to Month 60 for each model, and compare the results of the proposed model with G-O model, Vtub model and Teng & Pham model. The proposed model presents the smallest values of MSE(171.1531), AIC(280.1920) and PP(0.0517) in Table VI, which are far less than other models' values. Thus, we conclude clearly that the proposed model provides the best prediction power.

VI. CONCLUSIONS

In this paper, we propose a generalized software reliability model considering the uncertainty of field environments based on NHPP. Based on the general framework, several existing models are analyzed to be special cases of the generalized model and a new model is proposed with a three-parameter S-shaped fault-detection rate and a random environmental factor following the Weibull distribution. Comparisons of this model with several existing NHPP SRGMs have also been presented based on two real failure data sets. Six criteria have been used to compare the models in the context of LSE method. The results conclude that the proposed model can give a significantly improved goodness-of-fit and predictive power. Therefore, the results from the presented model are encouraging.

The limitations for the proposed model are analyzed as follows:

1. In our numerical experiments, quantity and type of the fault data sets seem to be limited. To be well-known, more data sets with more kinds may give much more effective validation for the model's ability. So we hope to apply this proposed model into more available failure data sets to evaluate the effectiveness of performance in a future research paper.
2. To simplify the model's mathematical calculation, the presented model assumes the environmental factor to be a one-dimensional random variant. But in practice, the environmental factor may be more complicated because it is affected by so many uncertain factors involved in the field environments. Thus it may take some kinds of complicated function forms, e.g., multidimensional random variant. Thus, more forms of the environmental factor should be studied in the future research.

APPENDIX

A. PROOF OF THE PROPOSED GENERALIZED MODEL THAT CONSIDERS THE UNCERTAINTY OF OPERATING ENVIRONMENTS WITH (7)

Given $t_0 = 0$, $m(t_0) = m(0) = 0$, then (6) can be simplified as follows:

$$m(t) = \int_0^{+\infty} e^{-xH(t)}(\int_0^t xa(\tau)h(\tau)e^{xH(\tau)}d\tau)g(x)dx \quad \text{(A.1)}$$

Therefore, according to (A.1) we can derive

$$\begin{aligned} m(t) &= \int_0^{+\infty} e^{-xH(t)}\left(\int_0^t xa(\tau)h(\tau)e^{xH(\tau)}d\tau\right)g(x)dx \\ &= \int_0^{+\infty} e^{-xH(t)}\left(\int_0^t a(\tau)de^{xH(\tau)}\right)g(x)dx \\ &= \int_0^{+\infty} e^{-xH(t)}\left(a(\tau)e^{xH(\tau)}|_0^t - \int_0^t e^{xH(\tau)}da(\tau)\right)g(x)dx \\ &= a(t)\int_0^{+\infty} f(x)dx - a(0)\int_0^{+\infty} e^{-xH(t)}g(x)dx \\ &\quad - \int_0^{+\infty} e^{-xH(t)}\left(\int_0^t e^{xH(\tau)}a'(\tau)d\tau\right)g(x)dx \\ &= a(t) - a(0)F^{(*)}(H(t)) \\ &\quad - \int_0^t a'(\tau)\left(\int_0^{+\infty} e^{-xH(t)}e^{xH(\tau)}g(x)dx\right)d\tau \end{aligned} \quad \text{(A.2)}$$

where $F^{(*)}(H(t))$ is the Laplace transform of the probability density function $g(x)$.

Then $e^{xH(\tau)}$ is in accordance with the Taylor formula expansion. Thus,

$$e^{xH(\tau)} = 1 + \frac{(xH(\tau))^1}{1!} + \cdots + \frac{(xH(\tau))^n}{n!} = \sum_{n=0}^{+\infty}\frac{x^nH^n(\tau)}{n!} \quad \text{(A.3)}$$

By substituting (A.3) into (A.2), we can obtain the following equation:

$$\begin{aligned} m(t) &= a(t) - a(0)F^{(*)}(H(t)) \\ &\quad - \int_0^t a'(\tau)\left(\int_0^{+\infty}\left(\sum_{n=0}^{+\infty}\frac{x^nH^n(\tau)}{n!}\right)e^{-xH(t)}g(x)dx\right)d\tau \\ &= a(t) - a(0)F^{(*)}(H(t)) \\ &\quad - \int_0^t a'(\tau)\sum_{n=0}^{+\infty}\frac{H^n(\tau)}{n!}\left(\int_0^{+\infty} x^ne^{-xH(t)}g(x)dx\right)d\tau \\ &= a(t) - a(0)F^{(*)}(H(t)) \\ &\quad - \int_0^t a'(\tau)\sum_{n=0}^{+\infty}\frac{H^n(\tau)}{n!}(-1)^nF^{(n)(*)}(H(t))d\tau \end{aligned} \quad \text{(A.4)}$$

where $F^{(n)(*)}(H(t))$ is the nth order differential of the Laplace transform of $g(x)$.

Thus, according to (A.4), we can derive

$$\begin{aligned} m(t) &= a(t) - a(0)F^{(*)}(H(t)) \\ &\quad - \int_0^t a'(\tau)\sum_{n=0}^{+\infty}\frac{H^n(\tau)}{n!}(-1)^nF^{(n)(*)}(H(t))d\tau \\ &= a(t) - a(0)F^{(*)}(H(t)) \\ &\quad - \sum_{n=0}^{+\infty}\frac{(-1)^n}{n!}F^{(n)(*)}(H(t))\int_0^t a'(\tau)H^n(\tau)d\tau \\ &= a(t) - a(0)F^{(*)}(H(t)) \\ &\quad - \sum_{n=0}^{+\infty}\frac{(-1)^n}{n!}F^{(n)(*)}(H(t))\int_0^t H^n(\tau)da(\tau) \\ &= a(t) - a(0)F^{(*)}(H(t)) \\ &\quad - \sum_{n=0}^{+\infty}\frac{(-1)^n}{n!}F^{(n)(*)}(H(t))[H^n(\tau)a(\tau)|_0^t \\ &\quad - \int_0^t a(\tau)nH^{n-1}(\tau)h(\tau)d\tau] \\ &= a(t) - a(0)F^{(*)}(H(t)) \\ &\quad - \sum_{n=0}^{+\infty}\frac{(-1)^n}{n!}F^{(n)(*)}(H(t))[a(t)H^n(t) \\ &\quad - n\int_0^t a(\tau)h(\tau)H^{n-1}(\tau)d\tau] \end{aligned} \quad \text{(A.5)}$$

This is just (7).

B. PROOF OF THE PROPOSED MODEL WITH (17)

Given $a(t) = N$, $h(t) = \frac{c}{1+ae^{-bt}}$, $g(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(\frac{x}{\lambda})^k}$ $(k, \lambda > 0; x \geq 0)$, so

$$H(t) = \int_0^t \frac{c}{1+ae^{-bu}}du = \frac{c}{b}\ln\left(\frac{a+e^{bt}}{1+a}\right) \quad \text{(B.1)}$$

Substituting all above assumptions into (7), we can have

$$\begin{aligned} m(t) &= N - NF^{(*)}(H(t)) \\ &\quad - N\sum_{n=0}^{+\infty}\frac{(-1)^n}{n!}F^{(n)(*)}(H(t)) \\ &\quad \times\left[H^n(t) - n\int_0^t h(\tau)H^{n-1}(\tau)d\tau\right] \end{aligned}$$

$$= N - NF^{(*)}(H(t)) - N\sum_{n=0}^{+\infty}\frac{(-1)^n}{n!}F^{(n)(*)}(H(t))\left[H^n(t) - \int_0^t dH^n(\tau)\right] = N\left(1 - F^{(*)}(H(t))\right) \quad \text{(B.2)}$$

where $F^{(*)}(H(t))$ is the Laplace transform of the Weibull pdf.

According to the definition of Laplace transform, $F^{(*)}(P) = \int_0^{+\infty} g(x)\mathrm{e}^{-Px}dx$, so for the Weibull distribution, we have

$$F^{(*)}(P) = \int_0^{+\infty}\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}\mathrm{e}^{-\left(\frac{x}{\lambda}\right)^k}\mathrm{e}^{-Px}dx \quad \text{(B.3)}$$

Then e^{-Px} is in accordance with the Taylor formula expansion. Thus,

$$e^{-Px} = 1 - Px + \cdots + (-1)^n\frac{(Px)^n}{n!} = \sum_{n=0}^{+\infty}\frac{(-1)^n P^n x^n}{n!} \quad \text{(B.4)}$$

By substituting (B.4) into (B.3), we can obtain

$$F^{(*)}(P) = \int_0^{+\infty}\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}\mathrm{e}^{-\left(\frac{x}{\lambda}\right)^k}\left(\sum_{n=0}^{+\infty}\frac{(-1)^n P^n x^n}{n!}\right)dx = \sum_{n=0}^{+\infty}\frac{(-1)^n\left(P^n\lambda^n k\right)}{n!}\int_0^{+\infty}\left(\frac{x}{\lambda}\right)^{k-1+n}\mathrm{e}^{-\left(\frac{x}{\lambda}\right)^k}d\left(\frac{x}{\lambda}\right) \quad \text{(B.5)}$$

According to the following formula $\int_0^{+\infty} y^{\frac{mn-2}{2}}\mathrm{e}^{-\rho y^n}dy = \frac{1}{|n|\rho^{\frac{m}{2}}}\Gamma\left(\frac{m}{2}\right)$, let $\frac{mn-2}{2} = k - 1 + n$, $\rho y^n = \left(\frac{x}{\lambda}\right)^k$, from (B.5), we can obtain

$$F^{(*)}(P) = \sum_{n=0}^{+\infty}\frac{(-1)^n\left(P^n\lambda^n k\right)}{n!}\cdot\frac{\Gamma\left(\frac{k+n}{k}\right)}{|k|} = \sum_{n=0}^{+\infty}\frac{(-1)^n\lambda^n}{n!}\Gamma\left(\frac{k+n}{k}\right)P^n. \quad \text{(B.6)}$$

Substituting (B.6) and (B.1) into (B.2), we can have

$$m(t) = N\left(1 - F^{(*)}(H(t))\right) = N\left(1 - \sum_{n=0}^{+\infty}\frac{(-1)^n\lambda^n}{n!}\Gamma\left(\frac{k+n}{k}\right)\left(\frac{c}{b}\ln\left(\frac{a+e^{bt}}{1+a}\right)\right)^n\right) \quad \text{(B.7)}$$

This is just (17).

ACKNOWLEDGMENT

The authors would like to thank the editor and anonymous reviewers for their valuable comments.

REFERENCES

[1] S. Yamada, *Software Reliability Modeling: Fundamentals and Applications*. Tokyo, Japan: Springer, 2014, pp. 1–20.
[2] H. Pham, *System Software Reliability*. London, U.K.: Springer, 2007, pp. 153–200.
[3] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Trans. Rel.*, vol. R-28, no. 3, pp. 206–211, Aug. 1979.
[4] P. N. Misra, "Software reliability analysis," *IBM Syst. J.*, vol. 22, no. 3, pp. 262–270, 1983.
[5] M. Ohba, "Software reliability analysis models," *IBM J. Res. Develop.*, vol. 28, no. 4, pp. 428–443, Jul. 1984.
[6] S. Yamada, H. Ohtera, and M. Ohba, "Testing-domain dependent software reliability models," *Comput. Math. Appl.*, vol. 24, pp. 79–86, Jul. 1992.
[7] H. Pham, *Software Reliability*. Singapore: Springer-Verlag, 1999, pp. 1–10.
[8] A. Wood, "Predicting software reliability," *Computer*, vol. 29, no. 11, pp. 69–77, Nov. 1996.
[9] A. L. Goel, "Software reliability models: Assumptions, limitations, and applicability," *IEEE Trans. Softw. Eng.*, vol. SE-11, no. 12, pp. 1411–1423, Dec. 1985.
[10] S. Yamada, K. Tokuno, and S. Osaki, "Imperfect debugging models with fault introduction rate for software reliability assessment," *Int. J. Syst. Sci.*, vol. 23, no. 12, pp. 2241–2252, Dec. 1992.
[11] J. Wang, Z. Wu, Y. Shu, and Z. Zhang, "An imperfect software debugging model considering log-logistic distribution fault content function," *J. Syst. Softw.*, vol. 100, pp. 167–181, Feb. 2015.
[12] X. M. Zhang, X. Teng, and H. Pham, "Considering fault removal efficiency in software reliability assessment," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 1, pp. 114–120, Jan. 2003.
[13] P. K. Kapur, D. Kumar, A. Gupta, and P. C. Jha, "On how to model software reliability growth in the presence of imperfect debugging and fault generation," in *Proc. 2nd Int. Conf. Rel. Saf. Eng.*, Jul. 2006, pp. 1–6.
[14] P. K. Kapur, H. Pham, S. Anand, and K. Yadav, "A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation," *IEEE Trans. Rel.*, vol. 60, no. 1, pp. 331–340, Mar. 2011.
[15] P. Roy, G. S. Mahapatra, and K. N. Dey, "An NHPP software reliability growth model with imperfect debugging and error generation," *Int. J. Rel., Qual. Saf. Eng.*, vol. 21, no. 2, 2014, Art. no. 1450008.
[16] K.-C. Chiu, Y.-S. Huang, and T.-Z. Lee, "A study of software reliability growth from the perspective of learning effects," *Rel. Eng. Syst. Saf.*, vol. 93, no. 10, pp. 1410–1421, Oct. 2008.
[17] P. Roy, G. S. Mahapatra, and K. N. Dey, "An S-shaped software reliability model with imperfect debugging and improved testing learning process," *Int. J. Rel. Saf.*, vol. 7, no. 4, pp. 372–387, 2013.
[18] Y. K. Malaiya, M. N. Li, J. M. Bieman, and R. Karcich, "Software reliability growth with test coverage," *IEEE Trans. Rel.*, vol. 51, no. 4, pp. 420–426, Dec. 2002.
[19] H. Pham and X. Zhang, "NHPP software reliability and cost models with testing coverage," *Eur. J. Oper. Res.*, vol. 145, no. 2, pp. 443–454, Mar. 2003.
[20] J.-Y. Park, G. Lee, and J.-H. Park, "A class of coverage growth functions and its practical application," *J. Korean Stat. Soc.*, vol. 1, pp. 1–7, Sep. 2008.
[21] C.-Y. Huang, S.-Y. Kuo, and M. R. Lyu, "An assessment of testing-effort dependent software reliability growth models," *IEEE Trans. Rel.*, vol. 56, no. 2, pp. 198–211, Jun. 2017.
[22] C.-Y. Huang, "Performance analysis of software reliability growth models with testing-effort and change-point," *J. Syst. Softw.*, vol. 76, no. 2, pp. 181–194, May 2005.
[23] X. Li, M. Xie, and S.-H. Ng, "Sensitivity analysis of release time of software reliability models incorporating testing effort with multiple change-points," *Appl. Math. Model.*, vol. 34, no. 11, pp. 3560–3570, 2010.
[24] N. Ahmad, M. G. M. Khan, and L. S. Rafi, "A study of testing-effort dependent inflection S-shaped software reliability growth models with imperfect debugging," *Int. J. Qual. Rel. Manage.*, vol. 27, no. 1, pp. 89–110, 2010.
[25] Q. Li, H. Li, and M. Lu, "Incorporating S-shaped testing-effort functions into NHPP software reliability model with imperfect debugging," *J. Syst. Eng. Electron.*, vol. 26, no. 1, pp. 190–207, Feb. 2015.
[26] C.-Y. Huang and C.-T. Lin, "Software reliability analysis by considering fault dependency and debugging time lag," *IEEE Trans. Rel.*, vol. 55, no. 3, pp. 436–450, Sep. 2006.

[27] Y. P. Wu, Q. P. Hu, M. Xie, and S. H. Ng, "Modeling and analysis of software fault detection and correction process by considering time dependency," *IEEE Trans. Rel.*, vol. 56, no. 4, pp. 629–642, Dec. 2007.

[28] H. Seheon and H. Pham, "Quasi-renewal time-delay fault-removal consideration in software reliability modeling," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 1, pp. 200–209, Jan. 2009.

[29] R. Peng, Y. F. Li, W. J. Zhang, and Q. P. Hu, "Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction," *Rel. Eng. Syst. Saf.*, vol. 126, pp. 37–43, Jun. 2014.

[30] B. Pachauri, J. Dhar, and A. Kumar, "Incorporating inflection S-shaped fault reduction factor to enhance software reliability growth," *Appl. Math. Modelling*, vol. 39, pp. 1463–1469, Mar. 2015.

[31] H. Okamura, T. Dohi, and S. Osaki, "A reliability assessment method for software products in operational phase–proposal of an accelerated life testing model," *Electron. Commun. Jpn.*, vol. 84, no. 8, pp. 25–33, Aug. 2001.

[32] H. Morita, K. Tokuno, and S. Yamada, "Markovian operational software reliability measurement based on accelerated life testing model," in *Proc. 11th ISSAT Int. Conf. Rel. Qual. Design*, Aug. 2005, pp. 204–208.

[33] K. Tokuno and S. Yamada, "User-oriented and -perceived software availability measurement and assessment with environmental factors," *J. Oper. Res. Soc. Jpn.*, vol. 50, no. 4, pp. 444–462, 2007.

[34] X. Teng and H. Pham, "Software cost model for quantifying the gain with considerations of random field environments," *IEEE Trans. Comput.*, vol. 53, no. 3, pp. 380–384, Mar. 2004.

[35] X. Teng and H. Pham, "A new methodology for predicting software reliability in the random field environments," *IEEE Trans. Rel.*, vol. 55, no. 3, pp. 458–468, Sep. 2006.

[36] H. Pham, "Mathematical systemability function approximations," in *Proc. 16th Int. Conf. Rel. Qual. Design*, 2010, pp. 6–10.

[37] H. Pham, "A new software reliability model with Vtub-shaped fault-detection rate and the uncertainty of operating environments," *Optimization*, vol. 63, no. 10, pp. 1481–1490, Oct. 2014.

[38] I. H. Chang, H. Pham, S. W. Lee, and K. Y. Song, "A testing-coverage software reliability model with the uncertainty of operating environments," *Int. J. Syst. Sci. Oper. Logistics*, vol. 4, no. 1, pp. 220–227, Oct. 2014.

[39] S. Inoue, J. Ikeda, and S. Yamada, "Bivariate change-point modeling for software reliability assessment with uncertainty of testing-environment factor," *Ann. Oper. Res.*, vol. 244, no. 1, pp. 209–220, Sep. 2016.

[40] K. Y. Song, I. H. Chang, and H. Pham, "A three-parameter fault-detection software reliability model with the uncertainty of operating environments," *J. Syst. Sci. Syst. Eng.*, vol. 26, no. 1, pp. 121–132, Feb. 2017.

[41] K. Y. Song, I. H. Chang, and H. Pham, "A software reliability model with a Weibull fault detection rate function subject to operating environments," *Appl. Sci.*, vol. 7, no. 10, p. 983, 2017.

[42] K. Y. Song, I. H. Chang, and H. Pham, "An NHPP software reliability model with S-shaped growth curve subject to random operating environments and optimal release time," *Appl. Sci.*, vol. 7, no. 12, p. 1304, 2017.

[43] K. Y. Song, I. H. Chang, and H. Pham, "Optimal release time and sensitivity analysis using a new NHPP software reliability model with probability of fault removal subject to operating environments," *Appl. Sci.*, vol. 8, no. 5, p. 714, May 2018.

[44] D. H. Lee, I. H. Chang, H. Pham, and K. Y. Song, "A software reliability model considering the syntax error in uncertainty environment, optimal release time, and sensitivity analysis," *Appl. Sci.*, vol. 8, no. 9, p. 1483, Sep. 2018.

[45] K. Y. Song, I. H. Chang, and H. Pham, "NHPP software reliability model with inflection factor of the fault detection rate considering the uncertainty of software operating environments and predictive analysis," *Symmetry*, vol. 11, p. 521, Sep. 2019.

[46] Q. Li and H. Pham, "NHPP software reliability model considering the uncertainty of operating environments with imperfect debugging and testing coverage," *Appl. Math. Model.*, vol. 51, pp. 68–85, Nov. 2017.

[47] H. Pham, L. Nordmann, and Z. Zhang, "A general imperfect-software-debugging model with S-shaped fault-detection rate," *IEEE Trans. Rel.*, vol. 48, no. 2, pp. 169–175, Jul. 1999.

[48] S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software error detection," *IEEE Trans. Rel.*, vol. R-32, no. 5, pp. 475–484, Dec. 1983.

[49] M. Ohba, "Inflection S-shaped software reliability growth model," in *Stochastic Models in Reliability Theory*, Berlin, Germany: Springer, 1984, pp. 144–162.

[50] S. A. Hossain and R. C. Dahiya, "Estimating the parameters of a non-homogeneous Poisson-process model for software reliability," *IEEE Trans. Rel.*, vol. 42, no. 4, pp. 604–612, Dec. 1993.

[51] H. Pham and X. Zhang, "An NHPP software reliability model and its comparison," *Int. J. Rel., Qual. Saf. Eng.*, vol. 14, no. 3, pp. 269–282, 1997.

[52] M. Zhao and M. Xie, "On maximum likelihood estimation for a general non-homogeneous Poisson process," *Scandin. J. Statist.*, vol. 23, no. 4, pp. 597–607, Dec. 1996.

[53] X. Zhang, D. R. Jeske, and H. Pham, "Calibrating software reliability models when the test environment does not match the user environment," *Appl. Stochastic Models Bus. Ind.*, vol. 18, no. 1, pp. 87–99, Jan. 2002.

[54] SourceForge.net. (2018). *An Open Source Software website*. [Online]. Available: http://sourceforge.net

[55] C. J. Hsu, C.-Y. Huang, and J.-R. Chang, "Enhancing software reliability modeling and prediction through the introduction of time-variable fault reduction factor," *Appl. Math. Model.*, vol. 35, no. 1, pp. 506–521, Jan. 2011.

[56] J. Xu and S. Yao, "Software reliability growth model with partial differential equation for various debugging processes," *Math. Problems Eng.*, vol. 6, pp. 1–13, Aug. 2016.

[57] X. Bao, M. Lu, and M. Cong, "Software reliability measurement," Softw. Dependability Res. Center, Beihang University, Beijing, China, Tech. Rep. 15, 2000.

QIUYING LI received the B.S. degree in mechanical design and manufacture and the M.S. degree in mechanics from Harbin Engineering University, Harbin, China, in 1996 and 1999, respectively, and the Ph.D. degree in software reliability from Beihang University, Beijing, China, in 2004, where she has been working, since 2004. From 2014 to 2015, she visited Rutgers University, Piscataway, NJ, USA, as a Visiting Scholar. She has authored or coauthored one book and more than 30 articles. Her current research interest includes software reliability theory and application.

HOANG PHAM (S'85–M'87–SM'92–F'05) received the B.S. degrees in mathematical statistics and computer science from Northeastern Illinois University, Chicago, IL, the M.S. degree in statistics from the University of Illinois Urbana–Champaign, Champaign, IL, USA, and the M.S. and Ph.D. degrees in industrial engineering from the State University of New York, Buffalo, NY, USA.

He is currently a Distinguished Professor with the Department of Industrial and Systems Engineering, Rutgers University, Piscataway, NJ, USA. Prior joining Rutgers, he was a Senior Engineering Specialist with Boeing Company, Seattle, WA, USA, and the Idaho National Engineering Laboratory, Idaho Falls, ID, USA. He has authored or coauthored more than 160 journal articles and has edited 12 books and 20 conference proceedings. He has also authored or coauthored six books: *Software Reliability*, *System Software Reliability*, *Reliability and Optimal Maintenance*, and *Maintenance for Industrial Systems*. His research interests include software reliability, system reliability modeling, maintenance, fault-tolerant computing, and complex-system modeling.

Dr. Pham is a Fellow of the Institute of Industrial Engineers (IIE). His numerous awards include the 2009 IEEE Reliability Society Engineer of the Year Award. He has been a Conference Chair and a Program Chair of more than 40 international conferences and workshops. He has been the Editor-in-Chief, an Editor, an Associate Editor, a Guest Editor, and a Board Member of many journals.