IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 June 2023, accepted 26 June 2023, date of publication 4 July 2023, date of current version 19 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3292301

RESEARCH ARTICLE

A Novel Software Reliability Growth Model Based on Generalized Imperfect Debugging NHPP Framework

HANG LUO[1], LIJIA XU[2], LIANG HE[1], LANDU JIANG[3], AND TING LONG[4]
[1]Department of Measurement and Control Engineering, School of Mechanical Engineering, Sichuan University, Chengdu 610065, China
[2]College of Mechanical and Electrical Engineering, Sichuan Agricultural University, Ya'an 625014, China
[3]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[4]School of Communication Engineering, Sichuan Post and Telecommunications College, Chengdu 610067, China

Corresponding author: Hang Luo (luohang@scu.edu.cn)

This work was supported in part by the New Century Higher Education Teaching Reform Project (the Ninth Phase) granted from Sichuan University under Grant SCU9162.

ABSTRACT Non-Homogeneous Poisson Process (NHPP) is a standard framework in the field of software reliability analysis. The core of NHPP consists in determining the Mean Value Function (MVF) of cumulative error number at a specific time slot. However, practice shows the difficulty in finding a general model to fit all sorts of fault data. A certain model is only sensitive to the specific object(s). Modeling failure MVF for NHPP still faces a number of challenges such as making reasonable explanation of assumption, determining fault detection rate per error, fault modification efficiency, error introduction rate, etc. In this research, we propose a novel Software Reliability Growth Model (SRGM) by leveraging generalized imperfect debugging NHPP framework. We first provide physical explanations for assumptions on error modification, error introduction and fault detection rate per error. Meanwhile, we generate a typical constraint relationship between the total error introduction rate and change rate of generalized residual errors. We also describe the fault detection rate per error with the form of exponential decay function, and use error reduction factor to form the new model. Furthermore, we make extensive discussions based on our proposed model. The experimental results confirm that our proposed model is effective on fault fitting and prediction, especially excellent on short-term prediction.

INDEX TERMS Software reliability growth, non-homogeneous Poisson process, error introduction rate, fault detection rate per error, error reduction factor, mean value function, maximum likelihood estimation.

NOMENCLATURE

Acronyms

NHPP	Non-Homogeneous Poisson Process.
MLE	Maximum Likelihood Estimation.
MVF	Mean Value Function.
AIC	Akaike's Information Criterion.
SSE	Sum of Squares for Error.
TBF	Time Between Failure.
CTBF	Cumulative Time Between Failure.
MTTF	Mean Time To Failure.
MTBF	Mean Time Between Failure.
SRGM	Software Reliability Growth Model.
LPETM	Logarithmic Poisson Execution Time Model.

Glossary

Software error	Inherent defect, such as wrong design, statement or omission, etc., is generated by programmer or designer.
Software fault	Abnormal external result, generated in program run, deviates from desired specification, and it is the result of software error.
Software failure	Unexpected output of software, the result of which can't match with required input value.

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

Notations

$N(t)$	Cumulative failure number found by time t, the distribution of which obeys NHPP with MVF $m(t)$.
$m(t)$	Expected failure number of software by time t, and $m(t) = E[N(t)]$.
$\lambda(t)$	Failure intensity by time t, and $\lambda(t) = m'(t)$.
$x(t)$	Number of modified errors by time t.
$a(t)$	Expected total error number of software by time t, the members of which consist of initial error(s) and introduced error(s) while correcting error.
$b(t)$	Fault detection rate per error by time t.
b	Initial fault detection rate per error.
α	Reduction parameter related to fault detection rate.
p	Error introduction ratio between error introduction rate and change rate of generalized residual errors.
q	Initial error reduction factor.
β	Error reduction parameter.
$R(\Delta t\|t)$	Software reliability function by time t for a mission time Δt. i.e., $R(\Delta t\|t) = P(\Delta T > \Delta t\|T = t)$ which denotes the probability of next failure time interval $\Delta T$is larger than given time interval Δtunder the condition of last failure time T is equal to given time t.
$\hat{}$	Estimated parameter or value.

I. INTRODUCTION

The NHPP, initially used for basic execution time model (also called as exponential type with Poisson distribution [1]), has gradually been a powerful research tool in the field of software reliability analysis since its usage to describe fault detection [2]. It is considered an essential metric due to its powerful fitting ability and excellent adaptability. Generally speaking, researchers or administrators in software reliability mainly focus on two key questions: (i) How long does it take to accumulate a certain number of faults? (ii) What is the reliability of software within a certain period? NHPP model can answer these two questions well.

The primary task of NHPP consists in determining its MVF (or intensity) of cumulative failure number at a particular time. In order to get the failure MVF, the fault data of a specific software object (module or codes) must be properly and effectively analyzed and fitted so as to get model's parameter(s). In the past decades, scholars have accomplished a number of valuable approaches on failure MVF study such as exponential model [2], [3], [4], S-shaped growth model [5], [6], [7], hyper-exponential growth model [8], [9], discrete reliability growth model [10], imperfect debugging model [8], generalized imperfect debugging model [11], [12], [13], [14], logistic testing-effort within imperfect debugging model [15], etc. TABLE 1 shows a summary of typical existing SRGMs based on NHPP framework.

TABLE 1. Summary of typical existing Srgms.

Model name	MVF	Comments
G-O[2]	$m(t) = a(1-e^{-bt})$ $a(t) = a$ $b(t) = b$	It comes from basic execution time model of exponential class [1].
Delayed S-shape (D-S)[6]	$m(t) = a[1-(1+bt)e^{-bt}]$ $a(t) = a$ $b(t) = \frac{b^2 t}{1+bt}$	Growth curve of observed cumulative failures is S-shape
Inflection S-shape (I-S)[5][26]	$m(t) = \frac{a(1-e^{-bt})}{1+\beta e^{-bt}}$ $a(t) = a$ $b(t) = \frac{b}{1+\beta e^{-bt}}$	Its fault detection rate per error contains a "learn" process with inflection factor β.
HD/G-O (H-D)[27]	$m(t) = \ln[e^a - c / (e^{ae^{-bt}} - c)]$	As a modified G-O model, its sufficient condition for the finite MLE is given.
Yamada Exponent (Y-E)[10]	$m(t) = a[1-e^{-r\alpha(1-e^{-\beta t})}]$ $a(t) = a$ $b(t) = r\alpha\beta e^{-\beta t}$	Attempt to consider testing-effort.
Yamada Rayleigh (Y-R)[10]	$m(t) = a[1-e^{-r\alpha(1-e^{-\beta t^2/2})}]$ $a(t) = a$ $b(t) = r\alpha\beta t e^{-\beta t^2/2}$	Attempt to consider testing-effort.
Yamada Imperfect #1 (Y-I-D1)[28]	$m(t) = \frac{ab}{\alpha+b}(e^{\alpha t} - e^{-bt})$ $a(t) = ae^{\alpha t}$ $b(t) = b$	Assume the total error number is an exponential function.
Yamada Imperfect #2 (Y-I-D2)[28]	$m(t) = a(1-e^{-bt})\left(1-\frac{\alpha}{b}\right) + \alpha a t$ $a(t) = a(1+\alpha t)$ $b(t) = b$	Assume the error introduction rate is a constant α.
P-N-Z[12]	$m(t) =$ $\frac{a}{1+\beta e^{-bt}}\left[(1-e^{-bt})\left(1-\frac{\alpha}{b}\right) + \alpha t\right]$ $a(t) = a(1+\alpha t)$ $b(t) = \frac{b}{1+\beta e^{-bt}}$	Assume the error introduction rate is a constant α, and the fault detection per error is a inflection S-shape function
H-K-L[15]	$m(t) = \frac{a}{1-\beta}\left[1-e^{-b(1-\beta)W^*(t)}\right]$ $a(t) = a+\beta m(t)$ $b(t) = b$ $W^*(t) = \frac{W}{1+Ae^{-\alpha t}} - \frac{W}{1+A}$	Assume the error introduction is proportional to the detected fault number, and the testing-effort is a logistic function.

However, practice shows that it's a great challenge finding the best or universal model for all the fault situations. In other words, a certain model is only sensitive to specific object(s), which limits the model's adaptability and generality. Although exploring diversified failure MVF on the basis of NHPP framework has become increasingly popular, developing models with wide range of adaptability still face great challenges in several aspects:

(i) Most of assumptions do not apply to all the situations. For example, the S-shaped models (i.e., inflection S-shaped model [5], [11], delayed S-shaped model [7] and connective type [16]) are the results of "learning" process which aims to improve the test efficiency dynamically. However, in actual test environments, the "learning" process has rarely been triggered due to limited available resources and non-operational profiles used for generating tests and business models [17].

(ii)Some assumptions are lack of reasonable and effective basis or authentic proof. Viewed from existing generalized imperfect debugging models [9], [11], [12], the total number of faults is almost assumed to be certain increasing functions of time. In addition, the error introduction rate is generally considered to be related to fault modification. How to describe such a relationship is still unclear.

(iii) Most of existing models are generally classified as the finite-fault type. Although the number of inherent errors hidden in software is limited due to finite codes, practice shows that the process of modification will introduce new errors in a certain probability [18]. Hence, it is possible that finite errors will cause infinite faults under the influence of modification. It's desirable to develop such an effective model which is an infinite-fault model while targeting finite-fault feature as well.

(iv) In NHPP framework, modeling failure MVF is the most important task. However, in most cases, explicit solution of failure MVF cannot be obtained except a small amount of special cases. Such limitation will bring great difficulty to estimate model's parameters. Therefore, the importance of exploring new model from which explicit solution of failure MVF can be easily obtained can never be overestimated.

To address such challenges, in this study, by taking the generalized imperfect debugging NHPP framework as the basis, we propose a novel SRGM. First of all, we construct typical constraint relation between error introduction rate and change rate of generalized residual errors based on reasonable assumptions which generally meet with error introduction process. Meanwhile, we propose exponential decay function to denote fault detection rate per error. Furthermore, we adopt error reduction factor to describe the effectiveness of error modification. Then, we derive explicit expression of failure MVF based on our proposed assumption. Moreover, we provide extensive discussions on failure MVF and deduce it to meet with traditional models under certain conditions. Experiments demonstrate that our proposed model has better performance in fault fitting and short-term prediction in certain range. Main contributions of this paper are summarized as follows:

(i) Differing from existing models, we attempt to take error reduction factor which is the form of exponential decay function of found error number as fault removal (modification) efficiency in generalized imperfect debugging framework.

(ii) Considering the change rate of residual errors which can be used for dynamically reflecting fault removal efficiency and for inferring the possibility of error introduction, we propose a new viewpoint: the error introduction rate is proportional to change rate of generalized residual errors.

(iii) Considering the fact that the fault detection efficiency is likely to be lower and lower in real testing environments, we take exponential decay function rather than traditional S-shaped curve as fault detection rate per error.

The rest of this paper is organized as follows. Firstly, NHPP framework and its related tasks are briefly introduced in Section II. Then, our proposed model, which includes presenting assumptions, making explanations and proving how to get model's result, is described in Section III. Next, in Section IV, five cases are provided with test results and discussions. In Section V, related work is reviewed and advantages of our approach are demonstrated. On the basis of verification, our findings are summarized and conclusions are drawn in Section VI. Finally, further researches are sketched in Section VII.

II. NHPP FRAMEWORK AND ITS RELATED TASKS

A. ANALYSIS OF FRAMEWORK BASED ON NHPP

As defined in Glossary, there are essential differences among the software error, fault and failure. In fact, the relationship among them is normally complex. For example, a single software error may lead to different faults, and a same fault may come from different errors, etc. In order to model a general model, it is necessary to simplify modeling conditions. So, in this paper, we assume each fault (or failure) corresponds to a sole error, and there is a one-to-one corresponding relation among them.

For a given software product, no matter whether it is in testing/application phase or not, the occurrences of faults related to inherent errors hidden in codes are random. Such randomness depends on the running environment/profile of software. Due to non-aftereffect property, Markov process can be used for describing the number of found faults and residual errors by time t. It is reasonable to assume the fault process of software mainly depends on the number of residual errors and operational profile [19].

Essentially speaking, the NHPP, which belongs to counting process, is based on following 4 assumptions:

(i) Failure process has the characteristic of independent increment, i.e., the number of found failures in time interval $(t, t + \triangle t]$ solely depends on current time t and interval length $\triangle t$.

(ii) In time interval $(t, t + \triangle t]$, the occurrence probability of one failure is $\lambda(t)\triangle t + o(\triangle t)$, i.e., $P\{N(t + \triangle t) - N(t) = 1\} = \lambda(t)\triangle t + o(\triangle t)$.

(iii) In time interval $(t, t + \triangle t]$, the occurrence probability of more than one failure is $o(\triangle t)$, i.e., $P\{N(t+\triangle t)-N(t) \geq 2\} = o(\triangle t)$.

(iv) At time $t = 0$, no failure occurs, i.e., $P\{N(0) = 0\} = 1$.

According to the Markov property, the probability of failure number by time $t + \triangle t$ is decided by the time interval $\triangle t$ and the found failure number by time t, namely, $P\{N(t + \triangle t) = j|N(t) = i\}$. Above four assumptions are applied to following full probability formula:

$$P\{N(t + \triangle t) = j\} = \sum_{i=1}^{\infty} P_{i,j}(t, \triangle t) \cdot P\{N(t) = i\}, \quad (1)$$

where, $P_{i,j}(t, \triangle t)$ denotes transfer function (i.e., conditional probability: $P\{N(t + \triangle t) = j|N(t) = i\}$). Then, a probability equation with recurrence relation is generated as follow:

$$P\{N(t + \triangle t) = n\} = [1 - \lambda(t)\triangle t] \cdot P\{N(t) = n\} + [\lambda(t)\triangle t] \cdot P\{N(t) = n - 1\} + o(\triangle t). \tag{2}$$

After solving Eq. (2) by means of using mathematical induction, the probability distribution of failure number by time t can be proved to be as follow:

$$P\{N(t) = n\} = \frac{[m(t)]^n}{n!} e^{-m(t)}, n = 0, 1, 2 \ldots \tag{3}$$

The distribution shown in Eq. (3) is exactly a NHPP distribution. Due to the property of independent increment, the reliability function of software can be derived from Eq. (3). Therefore, the reliability of software by time t for given mission time Δt can be simply expressed as follow:

$$R(\triangle t|t) = e^{-[m(t+\triangle t)-m(t)]}. \tag{4}$$

Eq. (4) produces a basic analysis framework for the occurrence of software failure. So long as the failure MVF $m(t)$ in this framework is obtained, some valuable information such as software reliability by time t for certain mission time t, number of residual errors by time Δt, necessary time t to reach a certain number of failures, etc., can be derived. Therefore, failure MVF $m(t)$ plays an important role in NHPP framework.

Generally, failure MVF $m(t)$ contains many unknown parameters. Hence, parameter estimation of failure MVF $m(t)$ is the primary task in software reliability analysis.

B. PARAMETER ESTIMATION BASED ON MLE

In statistics, MLE method has been widely used for parameter estimation. It has several excellent properties such as consistency, validation and asymptotic normality. For a given failure MVF $m(t)$ which satisfies NHPP, once enough failure data are collected during the testing process, the unknown parameters of $m(t)$ can be obtained by solving Logarithmic Likelihood Equation (LLE).

Suppose the cumulative number of failures detected in time interval $(0, t_i]$ is y_i, $i = 1, 2, \ldots, n$, and t_i satisfies a relation of $0 < t_1 < t_2 < \ldots < t_n$, then, by applying independent incremental property to NHPP, the corresponding Logarithmic Likelihood Function (LLF) can be expressed as follow:

$$\text{LLF} = \sum_{i=1}^{n} (y_i - y_{i-1}) \ln[m(t_i) - m(t_{i-1})] - m(t_n). \tag{5}$$

Suppose θ denotes an arbitrary unknown parameter in failure MVF $m(t)$, then, a typical form of likelihood equation obtained from Eq. (5) can be described as follow:

$$\sum_{i=1}^{n} \frac{\frac{\partial}{\partial\theta} m(t_i) - \frac{\partial}{\partial\theta} m(t_{i-1})}{m(t_i) - m(t_{i-1})} (y_i - y_{i-1}) - \frac{\partial}{\partial\theta} m(t_n) = 0. \tag{6}$$

Generally speaking, the LLE set from Eq. (6) contains complex non-linear equations, and their solutions can be obtained by means of making numerical calculation.

III. A NOVEL SOFTWARE RELIABILITY GROWTH MODEL

In NHPP framework, deriving traditional failure MVF is mainly based on four assumptions: (i) Each fault corresponds to a sole error; (ii) The fault rate per error remains the same; (iii) Once a fault is detected, modifying error immediately occurs, and error will be immediately removed; (iv) The fault detection rate is proportional to the number of residual errors hidden in software, and the proportionality coefficient is a constant. On the basis of these assumptions, the generalized imperfect debugging NHPP model is further based on the following three additional assumptions: (i) The above proportionality coefficient for the fault detection per error is generally considered as a function of time t, i.e. $b(t)$; (ii) Some new errors may be introduced in the process of modifying object error, in other words, the total error number $a(t)$ is a function of time t; (iii) Modifying error is not complete. It means each error detected by time t is modified with a certain probability. The generalized imperfect debugging NHPP model with error removal efficiency is denoted as follow [18]:

$$m'(t) = b(t)[a(t) - Qm(t)], \tag{7}$$

where, Q denotes error modification probability, and $Qm(t)$ denotes modified error number $x(t)$. For the sake of generality, our proposed new SRGM is mainly derived from following differential equation:

$$m'(t) = b(t)[a(t) - x(t)]. \tag{8}$$

Specifically, we are aware that the modified error number $x(t)$ is related to error reduction factor [20]. According to (8), we further model by means of developing new assumptions.

A. THREE IMPORTANT ASSUMPTIONS

1) EXPONENTIAL ERROR REDUCTION ASSUMPTION

In this assumption, we will use error reduction factor to describe differential constraint relationship between detected error (failure) number $m(t)$ and modified error number $x(t)$.

In actual debugging processes, modification efficiency may decrease with increase of detected faults (errors). This assumption is based on following three reasons, which result from reduction of correlation information among errors, mental inertia of current-task modification and instability of modification team: (i) Modifying error may be more and more difficult due to such possible situation: Error may be related to a certain part of removed fault set, and the remained parts with available information for modification may become decreasingly less with time. (ii) The inherent mental inertia of current-task modification may prevent debugger from further thinking about the relationship between errors. Such phenomena often lead debugger not to locate and modify other relevant errors according to modified error. (iii) Debuggers' mean modifying level and total experience may decrease for unpredictable reasons such as transfer, replacement and so on. (In many modern software companies, it is not unusual that experienced testers or debuggers are often transferred or replaced because of the need of engineering task. If successors do not take part in software design, their lack of experience may decrease the debugging efficiency in the early stage of change, so, the instability of modification team may make it less efficient on error debugging).

Since the modification efficiency may decrease with increase of detected errors, it is feasible that error reduction factor has the form of decreasing exponential function of $m(t)$. According to the definition of error reduction factor, the constraint relationship between detected error number $m(t)$ and modified error number $x(t)$ can be described as follow [20]:

$$\frac{x'(t)}{m'(t)} = qe^{-\beta m(t)}, \tag{9}$$

where, the parameter q is initial error reduction factor which denotes the ratio between initial error modification rate and initial error detection rate, and the parameter β denotes reduction parameter which determines the shape of declining curve. Therefore, Equation (9) has good expressive capability of error reduction. In this assumption, if $t \to 0$, $m(t)$ tends to be 0 and $x'(t)\big/m'(t)$ tends to be q.

2) ERROR INTRODUCTION RATE

In this section, we attempt to generate a constraint relationship between total error number $a(t)$ and modified error number $x(t)$, which is used for establishing an indirect connection with detected fault number $m(t)$.

Beyond all questions, the introduction of new error must generate during the period of error modifying phase rather than error detecting phase. Generally, no matter whether modifications are succeed or not during the period of error debugging, the more modification processes (note: for certain error, several-times debugging process may occur till error is completely modified), the larger the probability that new errors are introduced. Since the introduction of new errors is accompanied by error modification, it is reasonable that the change rate of residual errors is described by means of adopting dynamic process which contains introduction of new error and modification of error. So, the change rate of residual errors can't be simply denoted with $x'(t)$, but with $[a(t)-x(t)]'$(if the total error number is a constant, i.e. $a(t) = a_0$, then, $x'(t)$ and $[a(t) - x(t)]'$ essentially denote the same mathematical and physical meaning). We call the expression $[a(t) - x(t)]'$ as change rate of generalized residual errors, which is related to error modification intensity.

Considering the number of residual errors is generally a decreasing function of time t, we suppose the average ratio between the error introduction rate and the change rate of generalized residual errors is $-p$. The differential constraint relationship between total error number $a(t)$ and modified error number $x(t)$ can be shown as follow:

$$\frac{a'(t)}{[a(t) - x(t)]'} = -p. \tag{10}$$

Eq. (10) means the larger the generalized modification intensity, the larger the possibility of error introduction rate.

3) FAULT DETECTION RATE PER ERROR

In this section, we present an assumption that the fault detection rate per error is the form of exponential decay function of time t.

Strictly speaking, the fault detection rate of each residual error hidden in software is not the same. Like existing assumption, we also consider all the faults related to residual errors will be found with the same detection rate at a given time. Meanwhile, we believe that the fault detection rate per error may decrease with the time. The reasons can be explained as follows:

(i) As mentioned in Section I, in actual industrial and testing environments, so-called "learning" process seldom takes place due to limited resources and non-operational profiles.

(ii) Due to finite detecting conditions and methods together with limited testing experience, residual errors hidden in complex programs or abstract data structures are harder and harder exposed with time.

(iii) Fault detection efficiency may become lower and lower because long-term workload of recording, testing and analyzing execution process of program may cause the reliability of testers themselves to be lower and lower.

Considering the need of flexible expression, we assume that the fault detection rate per error obeys exponential decay function of time t, and the form of which is expressed as follow:

$$b(t) = be^{-\alpha t}, \tag{11}$$

where, the parameter b is initial fault detection rate per error, and the parameter α denotes reduction parameter.

B. A NOVEL SOFTWARE RELIABILITY GROWTH MODEL AND ITS FAILURE MVF

1) A NOVEL SOFTWARE RELIABILITY GROWTH MODEL

Based on the assumptions analyzed above, our proposed SRGM can be summarized as follows:

(i) Each fault (or failure) corresponds to a sole error;

(ii) Fault (or failure) of software obeys NHPP;

(iii) Once a fault (or failure) occurs, debugging (or modification) will be immediately executed;

(iv) Fault detection rate is proportional to the number of generalized residual errors hidden in software, and the ratio factor $b(t)$(i.e., fault detection rate per error) is an exponential decay function of time t;

(v) Fault removal efficiency is described with error reduction factor with the form of exponential decay function of detected fault number $m(t)$;

(vi) Error introduction rate is proportional to the change rate of generalized residual errors hidden in software, and the ratio factor is $-p$.

According to above assumptions, our proposed SRGM is expressed as following four constraint relationships:

$$\begin{cases} m'(t) = b(t)[a(t) - x(t)] & <i> \\ \dfrac{x'(t)}{m'(t)} = qe^{-\beta m(t)} & <ii> \\ \dfrac{a'(t)}{[a(t) - x(t)]'} = -p & <iii> \\ b(t) = be^{-\alpha t} & <iv>. \end{cases} \tag{12}$$

By jointly solving above four equations based on certain initial conditions, their corresponding solution (i.e., failure MVF $m(t)$) can be derived. Here, we give a Theorem to describe explicit failure MVF. The proof process is shown in Appendix.

Theorem 1: Suppose failure MVF, fault removal efficiency, error introduction rate and fault detection rate per error satisfy relationships shown as Eq. set (12), then, under the initial conditions $m(0) = 0$, $x(0) = 0$ and $a(0) = a_0$, the failure MVF is derived as follow:

$$m(t) = \frac{1}{\beta}\ln\left[\frac{\beta a_0 e^{[\beta a_0 - q(1-p)]\cdot\frac{b}{\alpha}(1-e^{-\alpha t})} - q(1-p)}{\beta a_0 - q(1-p)}\right]. \tag{13}$$

2) FURTHER DISCUSSIONS ABOUT FAILURE MVF

In this Section, we make further discussions about failure MVF, and give some deductive results:

(i) In (13), once the time t goes to infinity, then $m(t)$ will converge to upper bound, i.e., $\lim_{t\to\infty} m(t) = \frac{1}{\beta}\ln\left[\frac{\beta a_0 e^{[\beta a_0 - q(1-p)]\cdot\frac{b}{\alpha}} - q(1-p)}{\beta a_0 - q(1-p)}\right]$.

(ii) Under the assumed initial conditions $m(0) = 0$ and $x(0) = 0$, an identity $x(t) = \frac{q}{\beta}[1 - e^{-\beta m(t)}]$ can be followed by differentiation, directly from (12) < ii>. Substituting Eq. (13) into followed identify, we can get the number of modified errors shown in Eq.(14).

$$x(t) = \frac{q}{\beta}\left[1 - \frac{\beta a_0 - q(1-p)}{\beta a_0 e^{[\beta a_0 - q(1-p)]\cdot\frac{b}{\alpha}(1-e^{-\alpha t})} - q(1-p)}\right]. \tag{14}$$

(iii) Under the initial condition $a(0) = a_0$, substituting Eq. (14) into (12) <iii>, we can get the total error number:

$$a(t) = a_0 + px(t) = a_0 + \frac{pq}{\beta}\left[1 - \frac{\beta a_0 - q(1-p)}{\beta a_0 e^{[\beta a_0 - q(1-p)]\cdot\frac{b}{\alpha}(1-e^{-\alpha t})} - q(1-p)}\right]. \tag{15}$$

(iv) If $\alpha \to 0$(it means the fault detection rate per error approximately equals to a constant), then, Eq. (13) can be rewritten as:

$$m(t) = \frac{1}{\beta}\ln\left[\frac{\beta a_0 e^{[\beta a_0 - q(1-p)]\cdot bt} - q(1-p)}{\beta a_0 - q(1-p)}\right]. \tag{16}$$

(v) In Eq. (16), if $p \to 0$(it means error introduction rate is also approximately close to 0), then, Eq. (16) can be rewritten as:

$$m(t) = \frac{1}{\beta}\ln\left[\frac{\beta a_0 e^{(\beta a_0 - q)\cdot bt} - q}{\beta a_0 - q}\right]. \tag{17}$$

(vi) Under the conditions of $\alpha \to 0$and $p \to 0$, the limit $\beta a_0 \to q$ is satisfied based on (12) <ii >. So, in Eq. (17), if $b \to 1$, $\beta a_0 \to q$, we can obtain a general expression:

$$m(t) = \frac{1}{\beta}\ln(1 + \beta a_0 t). \tag{18}$$

Eq. (18) is just the Logarithmic Poisson Execution Time Model (LPETM) [21]. In this sense, our proposed model derived from Eq. set (12), which contains novel viewpoints, can be considered as a generalized form of LPETM. Essentially speaking, Eq. (18) is an infinite fault model. However, by consideration of error introduction, error reduction factor and decrease of fault detection, the derived new model Eq. (13), which combines concave model with infinite fault model, has characteristic of infinite fault while converges to an upper bound.

IV. TESTING AND DISCUSSION

Viewed from application, whether a model is good or not can be generally determined by a variety of comparisons. In this section, we will make test and make analysis based on some typical evaluation criteria.

A. EVALUATION CRITERIA OF MODEL

For a model, the fitting capability and predictive power are normally evaluated by comparing the values of some criteria. As a typical statistical parameter, SSE criterion, used for calculating the square sum of the errors of the corresponding points between the fitting data and the original data, is described as follow:

$$\mathrm{SSE} = \sum_{j=1}^{k}\sum_{i=1}^{n}[y_{ij} - \hat{m}_j(t_i)]^2, \tag{19}$$

where, y_{ij} is the actual fault number of j^{th} type of fault data by time t_i, and $\hat{m}_j(t_i)$ denotes the estimated total fault number of j^{th} type of fault data by time t_i. Our proposed model contains solely a type, hence, $k = 1$. The power of maximizing likelihood function of model may be evaluated by using AIC criterion shown as follow [22]:

$$\mathrm{AIC} = -2\ln(LF_{\max}) + 2N, \tag{20}$$

where, $\mathrm{LF}_{\max}$ denotes maximum of logarithmic likelihood function, and N denotes the parameter number of model.

With larger penalty term than that of AIC, another criteria BIC, used for prevent model's high complexity for high precision of fitting, is defined as follow [23]:

$$\mathrm{BIC} = -2\ln(\mathrm{LF}_{\max}) + N\ln(n), \tag{21}$$

where, n denotes the number of sample.

In addition, R-square [24], taken as a correlation index of regression curve equation, is defined as follow:

$$\mathrm{R}^2 = 1 - \sum_{i=1}^{n}[\hat{m}(t_i) - m_i]^2 / \sum_{i=1}^{n}(m_i - m_{ave})^2, \tag{22}$$

where, m_i is the detected number of faults by time t_i, $\hat{m}(t_i)$ is the estimated number of faults by time t_i, and m_{ave} is the average value of m_i.The value of R-square is limited in [0, 1].

Generally, for a given data set, the smaller the obtained values of SSE and AIC are, the higher the fitting degree is. The larger the value of R-square is, the better the model fits.

B. TEST RESULTS OF SEVERAL CASES

In this Section, we use five public data sets to illustrate the effectiveness of our proposed model. Firstly, we select the first part data from the data sets to fit model by using MLE method. Secondly, taking generated model parameters as basis, we use failure MVF to predict subsequent value. Finally, on the basis of making comparisons of fitting, prediction and reliability performance on each case, we assess the quality of new model.

1) FAILURE DATA FROM THE SOFTWARE PRODUCT OF TANDEM COMPUTER COMPANY (CASE 1)

We consider such a fault data set [25], which is from the 1^{st} version of certain software product published by Tandem Computer Company. In order to make results to be comparable, we select the first 9 samples in data set to estimate the parameters of model. Then, we use proposed model with estimated parameters to predict expected number of residual errors hidden in software at a specific time which corresponds to an actual error number. Furthermore, we also give some reliability predictions under several mission intervals and make reliability comparison among several models under specific mission time.

According to estimated parameters shown in TABLE 2, we make some explanations: (i) Modifying (or Debugging) is imperfect. The initial error reduction factor (i.e., $\hat{q} = 0.9969$) means about 1 errors are not modified in every 100 detected faults per hour, and as time goes on, the error reduction factor decreases with reduction parameter 0.02271; (ii) During the period of modification, new errors are introduced. Its corresponding ratio (i.e., $\hat{p} = 0.08804$) between the error introduction rate and the change rate of generalized residual errors means 9 or so errors are introduced in every 100 modified residual errors (note: according deduced model, the introduced error number, i.e., $a(t) - a_0$, is equivalent to the result of converted $x(t)$ in a ratio $p/(1+p)$); (iii) The initial fault detection rate per error (i.e., $\hat{b} = 1.9541 \times 10^{-4}$) means about 2 faults are detected per hour in every 10000 residual errors hidden in software. The fault detection rate per error declines slowly with attenuation constant $\hat{\alpha} = 1.5099 \times 10^{-4}$.

TABLE 2. Mle solutions based on the first 9 samples in fault data set [25].

PARM	$\hat{a}_0$	$\hat{p}$	$\hat{q}$	$\hat{b}$	$\hat{\alpha}$	$\hat{\beta}$
Value	119.23	0.08804	0.9969	1.9541×10^{-4}	1.5099×10^{-4}	2.2711×10^{-2}

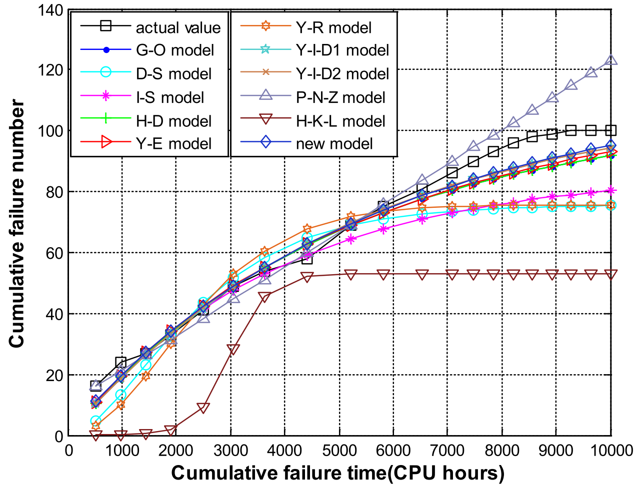

FIGURE 1. Fitting and predictive curves of several models based on the first 9 fault data in fault data set [25].

On the basis of estimated parameters listed in TABLE 2, and by means of using Eq. (13), we give fitting and prediction values of proposed model together with several existing models, which of them are listed in TABLE 3, and accordingly, several fitting and prediction curves are shown in FIG.1

Viewed from the values of SSE and R listed in TABLE 3, new model performs well in aspect of fitting and prediction (Note: except being inferior to P-N-Z model in aspect of fitting, new model has the best fitting and prediction capacity). Though the P-N-Z model performs excellently in aspect of fitting on the first segment of this data set, its predictive ability in later period is obviously inferior to new model. The main reason consists in such a fact that the fault number of P-N-Z model is almost a linear growth function of time while new model will converge gradually if enough time is considered.

Although the predictive value obtained from new model is generally smaller than the real value, new model itself has better work performance on this data set. Some analyses together with conclusions are given as follows: (i) At the end of testing phase, the final prediction value of new model is closest to the actual value. (ii) New model has better predictive power in the subsequent application phase. The expected error number predicted from new model will be 100 by cumulative testing time $t = 11489$ (CPU hours), and the expected error number will eventually converge to 118.7883(i.e., 119 or so). (iii) According to Eq. (15), the total error number will be 122.3708 (123 or so) by the end of test phase (i.e., $t = 10000$ CPU hours). This means 4 or so new errors are introduced

TABLE 3. Fitting and predictive values obtained from several models based on fault data set [25].

Testing time(week)	Cumulative testing time (CPU hours)	Found fault number	G-O model	D-S model	I-S model	H-D model	Y-E model	Y-R model	Y-I-D1 model	Y-I-D2 model	P-N-Z model	H-K-L model	new model
1	519	16	10.56	4.73	10.98	10.56	10.95	3.02	10.41	10.42	16.01	0.03	11.16
2	968	24	18.85	13.18	19.36	18.84	19.39	9.86	18.61	18.63	21.07	0.13	19.59
3	1430	27	26.62	23.09	27.00	26.61	27.21	19.48	26.36	26.38	26.28	0.51	27.32
4	1893	33	33.70	32.72	33.80	33.70	34.27	30.07	33.46	33.48	31.50	1.89	34.28
5	2490	41	41.91	43.58	41.43	41.91	42.36	42.90	41.75	41.76	38.24	9.26	42.31
6	3058	49	48.85	51.92	47.66	48.85	49.19	52.95	48.80	48.81	44.64	28.43	49.12
7	3625	54	55.02	58.38	53.03	55.02	55.24	60.53	55.12	55.13	51.04	45.50	55.22
8	4422	58	62.58	64.83	59.35	62.58	62.66	67.60	62.94	62.94	60.03	52.25	62.82
9	5218	69	69.00	69.00	64.48	69.00	69.02	71.64	69.66	69.65	69.00	53.08	69.42
10	5823	75	73.23	71.10	67.72	73.23	73.25	73.39	74.13	74.12	75.82	53.17	73.88
11	6539	81	77.61	72.77	70.95	77.62	77.70	75.55	78.81	78.80	83.90	53.18	78.62
12	7083	86	80.54	73.62	73.03	80.54	80.73	75.03	81.97	81.96	90.04	53.18	81.87
13	7487	90	82.51	74.09	74.40	82.52	82.80	75.24	84.12	84.11	94.60	53.18	84.10
14	7846	93	84.13	74.41	75.49	84.14	84.52	75.36	85.90	85.89	98.64	53.18	85.97
15	8205	96	85.64	74.70	76.49	85.65	86.15	75.44	87.56	87.56	102.69	53.18	87.73
16	8564	98	87.04	74.87	77.40	87.05	87.68	75.49	89.12	89.11	106.74	53.18	89.40
17	8923	99	88.35	75.03	78.23	88.36	89.12	75.52	90.58	90.58	110.79	53.18	90.98
18	9282	100	89.56	75.16	78.99	89.60	90.49	75.54	91.94	91.95	114.84	53.18	92.47
19	9641	100	90.69	75.26	79.67	90.70	91.77	75.55	93.22	93.23	118.89	53.18	93.87
20	10000	100	91.73	75.34	80.30	91.75	92.99	75.56	94.42	94.44	122.94	53.18	95.21
AIC			-143.0	-119.9	-140.2	-141.0	-141.4	-103.1	-141.0	-141.0	-143.1	148.2	-135.9
BIC			-142.6	-119.5	-139.6	-140.4	-140.8	-102.5	-140.4	-140.4	-142.3	149.4	-134.7
SSE(fitting)			79.6	340.7	72.6	79.6	73.6	594.8	87.6	87.4	50.8	4283.6	71.2
SSE(prediction)			784.3	4236.8	3351.4	784.0	673.8	3967.6	470.1	469.9	1441.9	17783	432.2
R^2			0.9679	0.8626	0.9707	0.9679	0.9703	0.7601	0.9647	0.9648	0.9795	-0.728	0.9713

during the debugging process. According to the records of user operational phase, about 20 new errors were detected [25]. This means the final expected error number (i.e., 119 or so) predicted from new model meets well with the real error number to a greater degree.

Viewed from FIG.1 and TABLE 3, the fitting abilities of listed models are relatively close except model D-S, Y-R and H-K-L. The exceptions show these three models have certain feature of s-shape which makes their corresponding models not strictly be reliability growth models, and such conclusion can be further verified from angle of reliability shown in FIG.2 (b). The prediction value of P-N-Z model is obviously larger than actual value while that of other models are opposite. Moreover, the prediction values of model D-S and Y-R are much less in the case of underestimation. The fitting and prediction capabilities of model I-S and H-K-L are totally not good.

On the basis of Eq. (4), we show several reliability growth curves in FIG. 2 (a) to make further analysis. Since the MTBF is 500 (CPU hours), so, we select 50, 100, 200 and 500 as mission time in turn. By comparing those curves, we can draw following conclusions: (i) The reliability of current software version increases gradually with test time. The development trend of curves meets with the fact that the number of found new faults in the test interval is generally getting less and less; (ii) Given different mission time, the reliability at certain time point is not the same. The longer the mission time, the lower the corresponding reliability; (iii) Viewed from last data of each curve, the reliability of current software version may not be optimistic. The reason consists in such a fact: For the first

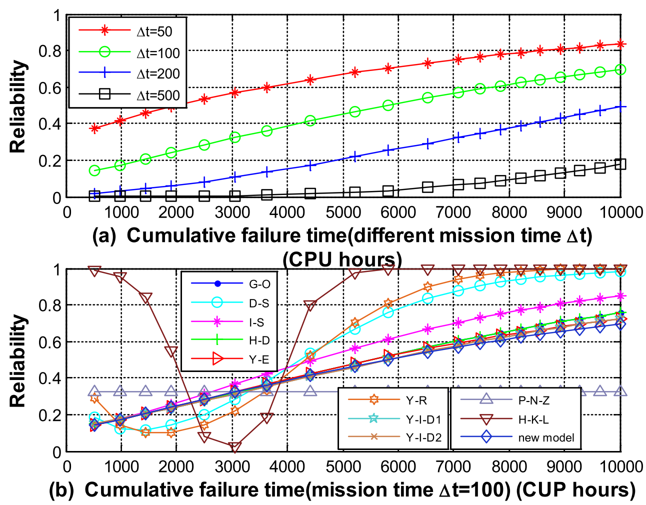

FIGURE 2. Reliability curves based on proposed model (a), and on several models under mission time Δt=100 CPU hours (b).

curve (marked with asterisk), its reliability is comparatively high in the later stage of test, but its mission time is shorter than that of other cases. Although the mission time of last curve (marked with square) is longer, its reliability in later stage of test is not satisfactory. The number of new faults found in user operational stage can verify this conclusion.

In FIG.2 (b), the reliability curves of model D-S, Y-R and H-K-L decrease at the beginning of testing phase while increase to 1 with different rate in late stages of testing phase. In addition, the reliability growth curve of model P-N-Z almost keeps a horizontal line. Viewed from those

TABLE 4. Mle solutions based on the first 26 samples in fault data set [2].

PARM	$\hat{a}_0$	$\hat{p}$	$\hat{q}$	$\hat{b}$	$\hat{\alpha}$	$\hat{\beta}$
Value	64.7091	0.03676	0.68939	0.00343	0.00603	0.03947

TABLE 5. Comparison of goodness-of-fit and predictive power based on fault data set [2].

Model	SSE(fitting) $\sum_{k=1}^{26}[y_k - \hat{m}(t_k)]^2$	SSE(prediction) $\sum_{k=27}^{34}[y_k - \hat{m}(t_k)]^2$	AIC	BIC	R^2
G-O	120.53	13.04	86.25	88.76	0.9176
D-S	39.97	137.97	83.84	86.36	0.9727
I-S	47.31	159.56	86.31	90.09	0.9677
H-D	120.47	12.91	88.25	92.02	0.9176
Y-E	120.87	11.95	88.26	92.03	0.9174
Y-R	26.37	151.07	86.39	90.16	0.9820
Y-I-D1	121.55	15.73	88.25	92.02	0.9169
Y-I-D2	121.55	15.73	88.25	92.02	0.9169
P-N-Z	17.17	354.95	83.01	88.05	0.9883
H-K-L	91.64	196.40	96.15	103.7	0.9373
new model	126.38	9.18	94.49	102.0	0.9136

inconceivable phenomena, it seems that the corresponding modes with abnormal reliability are not suitable for this failure data set. On the contrary, the reliability of new model increases gradually from relatively small value. It means proposed model tends to be conservative in reliability prediction.

2) FAILURE DATA FROM NTDS (CASE 2)

We select the Naval Tactical Data System (NTDS) [2] whose lifecycle was divided into four phases, i.e., production phase (26 data), testing phase (5 data), usage phase (1 data) and follow-up testing phase (2 data) as research subjects.

Considering the integrity of data found in production phase, we select the first 26 samples in data-set to fit new model together with some existing models, and then predict next 8 data. The estimated parameters are listed in TABLE 4 and the fitting and prediction results (viewed from SEE) in TABLE 5.

Several fitting and prediction curves are shown in FIG. 3 (note: the last 8 values of each curve are the prediction values) and reliability curves shown in FIG. 4.

Viewed from SSE, AIC (BIC) and R^2 shown in TABLE 5, it seems that the capability of proposed model in aspect of fitting is not good. However, the short-term predictive capability of proposed model is better than that of other models on this data set. Although the P-N-Z model shows excellent fitting characteristic, it has poor prediction ability.

Obviously, this failure data set itself has s-shape feature which can be used for checking a given model whether is an s-shape model or not by means of fitting method. The model D-S, I-S, Y-R and H-K-L has relatively good performance in aspect of fitting. So, it shows that the mentioned above 4 models are s-shape models, and this conclusion matches well with the conclusion drawn in case 1.

In FIG. 4(a), the reliability of this software starts growing from time $t = 116$ or so, and its growth rate is relatively fast

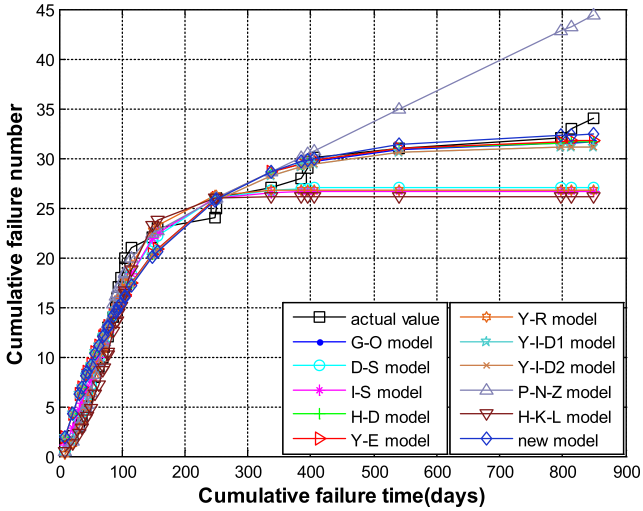

FIGURE 3. Fitting and predictive curves of several models based on the first 26 fault data in failure data set [2].

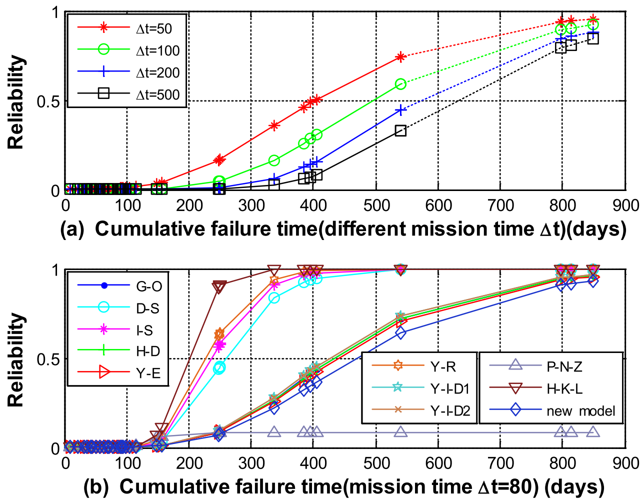

FIGURE 4. Reliability curves based on proposed model (a), and on several models under mission time Δt=80 days (b).

from time $t = 249$ or so. At the end of testing phase (i.e. $t = 540$), the reliability of this software is promising because the reliability values for different mission time are all not too low. In fact, the subsequent four curves (marked with dashed line) which correspond to the last three data in data set get closer and closer. At the last time point, the reliabilities of four curves are all over 0.9. It's not hard to conclude that this software was better modified at last. Above conclusions can be effectively verified from test records due to the fact that the value of TBF in the later stage is generally longer than that in the early stage.

In FIG.4 (b), new model is similar with G-O, H-D, Y-I-D1, Y-I-D2 and Y-E model in aspect of reliability growth trend. As in case 1, P-N-Z model performs poor reliability prediction ability for its small and almost invariant results from time $t = 156$ or so. On the contrary, the reliability prediction values of D-S, I-S, Y-R and H-K-L model grow so fast that

TABLE 6. Mle solutions based on the first 15 samples in fault data set [29].

PARM	$\hat{a}_0$	$\hat{p}$	$\hat{q}$	$\hat{b}$	$\hat{\alpha}$	$\hat{\beta}$
Value	74.5899	0.04614	0.98990	1.3054×10^{-3}	3.5569×10^{-3}	4.1844×10^{-5}

TABLE 7. Comparison of goodness-of-fit and predictive power on the first 15 samples in data set [29].

Model	SSE(fitting) $\sum_{k=1}^{15}[y_k - \hat{m}(t_k)]^2$	SSE(prediction) $\sum_{k=16}^{22}[y_k - \hat{m}(t_k)]^2$	AIC	BIC	R^2
G-O	24.15	5.15	48.44	49.86	0.9137
D-S	25.73	83.89	54.67	56.01	0.9081
I-S	22.56	8.78	50.46	52.59	0.9194
H-D	24.16	5.16	50.44	52.56	0.9137
Y-E	23.65	7.13	50.49	52.62	0.9155
Y-R	17.68	96.41	58.48	60.60	0.9368
Y-I-D1	24.98	8.56	50.45	52.57	0.9108
Y-I-D2	24.98	8.56	50.45	52.57	0.9108
P-N-Z	46.25	232.86	57.10	59.94	0.8348
H-K-L	18.57	111.67	57.81	62.06	0.9337
new model	24.14	4.64	56.47	60.72	0.9138

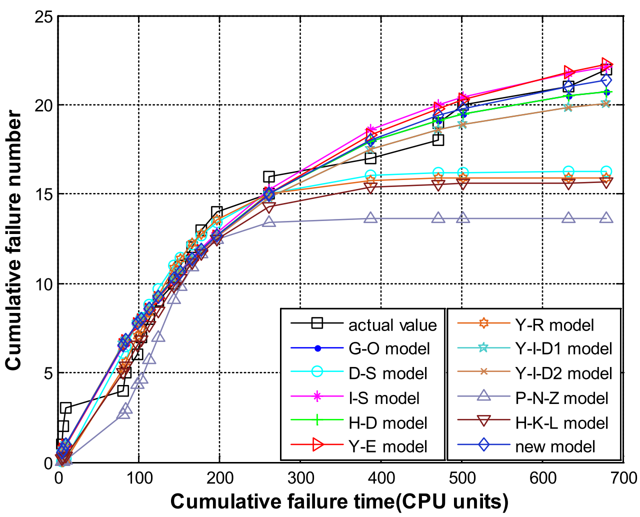

FIGURE 5. Fitting and predictive curves of several models based on the first 15 fault data in fault data set [29].

their values increase to 1 soon. Obviously, the reliability performances of last five models mentioned above are far from actual case.

3) FAILURE DATA FROM SYSTEM T AT AT & T (CASE 3)

In this section, we test the fitting and predictive capability of new model on the released failure data from System T at AT&T [29]. We select the first 15 of 22 found fault data for fitting, then use subsequent 7 data for assessing model's predictive capability. Estimated parameters are shown in TABLE 6, and fitting and prediction curves are shown in FIG. 5.

TABLE 8. Running values of several physical quantities from new model based on fault data set [29].

Fault number	Cumulative TBF(CPU units)	$\hat{b}(t)$	$\hat{a}(t)$	$\hat{m}(t)$	$\hat{x}(t)$	$\frac{\hat{x}'(t)}{\hat{m}'(t)}$
0	0	0.0013	74.59	0	0	0.9899
1	5.50	0.0013	74.61	0.53	0.52	0.9899
2	7.33	0.0013	74.62	0.70	0.69	0.9899
3	10.08	0.0013	74.63	0.96	0.95	0.9899
4	80.97	0.0010	74.88	6.56	6.49	0.9896
5	84.91	0.0010	74.88	6.82	6.75	0.9896
6	99.89	0.0009	74.93	7.78	7.70	0.9896
7	103.36	0.0009	74.94	7.99	7.91	0.9896
8	113.32	0.0009	74.96	8.58	8.49	0.9895
9	124.71	0.0008	74.99	9.22	9.13	0.9895
10	144.59	0.0008	75.04	10.27	10.17	0.9895
11	152.40	0.0008	75.06	10.66	10.55	0.9895
12	166.99	0.0007	75.09	11.35	11.24	0.9894
13	178.41	0.0007	75.11	11.88	11.74	0.9894
14	197.35	0.0006	75.14	12.67	12.54	0.9894
15	262.65	0.0005	75.24	14.99	14.83	0.9893
16	262.69	0.0005	75.24	14.99	14.83	0.9893
17	388.36	0.0003	75.38	18.05	17.86	0.9892
18	471.05	0.0002	75.44	19.39	19.18	0.9891
19	471.50	0.0002	75.44	19.39	19.19	0.9891
20	503.11	0.0002	75.45	19.80	19.59	0.9891
21	632.42	0.0001	75.51	21.05	20.83	0.9890
22	680.02	0.0001	75.52	21.38	21.16	0.9890

Several values of goodness-of-fit are listed in TABLE 7. Viewed from AIC (BIC), SSE and R^2, the fitting quality of new model is in middle level while the prediction capability of new model is superior to other models.

Taking estimated parameters as an example, we give some information shown in TABLE 8. According to listed items and their corresponding contents, we make analysis and explanations: (i) The reason why error reduction factor (i.e., $x'(t)\big/m'(t)$) almost keep a relatively large value lies in such a fact that it decreases from a not small initial value $\hat{q}$ and small attenuation parameter $\hat{\beta}$; (ii) The fault detection rate per error (i.e., $b(t)$) also drops slowly to keep a relatively large value for its small attenuation parameter $\hat{\alpha}$; (iii) Final fault number $\hat{m}(t)$ (i.e. 21 or so, which is rounded from 21.38) and modified error number $\hat{x}(t)$ (i.e. 21, rounded from 21.16) almost match with found fault number (i.e. 22); (iv) Almost no new error is introduced due to small error introduction ratio and too small number of unmodified errors in a short test time. So, we further draw two conclusions: (i) Error modification on this software is satisfactory, which is based on a common sense: the larger error reduction factor, the higher error modification efficiency. After all, the error reduction factor can be approximately expressed in another way, i.e., $[\Delta x(t)]\big/[\Delta m(t)]$ within a certain time interval Δt. Hence, the larger value of $[\Delta x(t)]\big/[\Delta m(t)]$ means the increment in number of modified errors accounts for a larger ratio of the increment in number of found faults; (ii) The large error reduction factor together with high fault detection rate per error means reliability growth of this software is robust and satisfactory.

Considering the MTBF is about 30.91 CPU units, we select 10, 30, 50 and 70 CPU units as mission time respectively

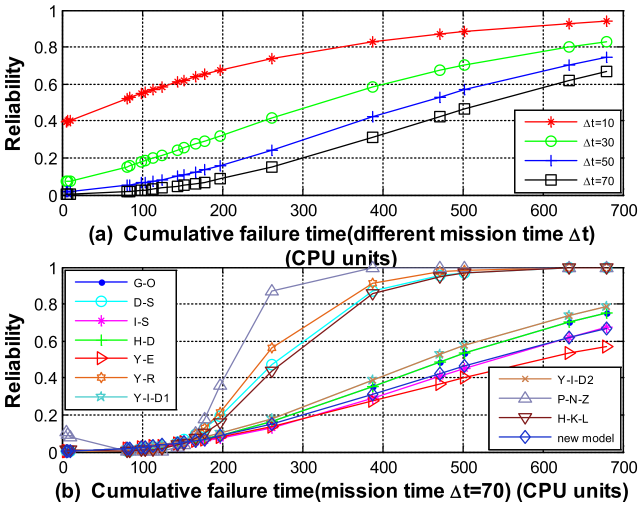

FIGURE 6. Reliability curves based on proposed model (a), and on several models under mission time Δt=70 CPU units (b).

to assess reliability quality of models. Several reliability growth curves based on the estimated parameters from the first 15 data are shown in FIG. 6(a). The result that the steady growing curves get closer and closer and their final values are all much high means the reliability growth of this software is promising. Viewed from FIG. 6(b), new model together with model G-O, H-D, Y-E, Y-I-D1 and Y-I-D2 shows excellent quality of lasting steady-state growth in aspect of reliability prediction. On the contrary, the reliabilities of model D-S, Y-R, P-N-Z and H-K-L, similar to that in case 1 and 2, also show abnormal behaviors (i.e., their values decrease to close to 0 in early stage and then get to 1 with a sharp increase rate in middle stage). Such phenomena show the mentioned above four models have obvious s-shape features, so, these modes do not fit well on this data set. Viewed from the results (at least from angle of predictions) shown in TABLE 7, this conclusion can be generally verified.

4) FAILURE DATA FROM IEEE STD 1633™-2016 (CASE 4)

In this section, a typical fault data set from IEEE Std 1633™-2016 [30] is applied to check the effectiveness of new model's fitting and predictive capability. In fault data set, the fault time between failures is not known, and only the number of faults and their test hours are given per day. So, we fit cumulative fault numbers which correspond to their cumulative test hours. In our verification, the first 21 of 29 fault data are used for fitting, and last 8 of 29 fault data for prediction. Similarly, considering the TBF of fault data set is 24.9524, we give several models' reliability levels, selecting several adjacent values of TBF to make comparison.

The estimated model parameters based on the first 21 fault data are shown in TABLE 9, and the fitting and predictive curves and reliability levels are shown in FIG. 7 and FIG.8 respectively. Fitting and predictive values and corresponding AIC (BIC), SSE and R^2 are listed in TABLE 10.

TABLE 9. Mle solutions based on the first 21 samples in fault data set [30].

PARM	$\hat{a}_0$	$\hat{p}$	$\hat{q}$	$\hat{b}$	$\hat{\alpha}$	$\hat{\beta}$
Value	208.660	0.06953	0.82165	9.5868×10^{-4}	2.7942×10^{-3}	1.5784×10^{-2}

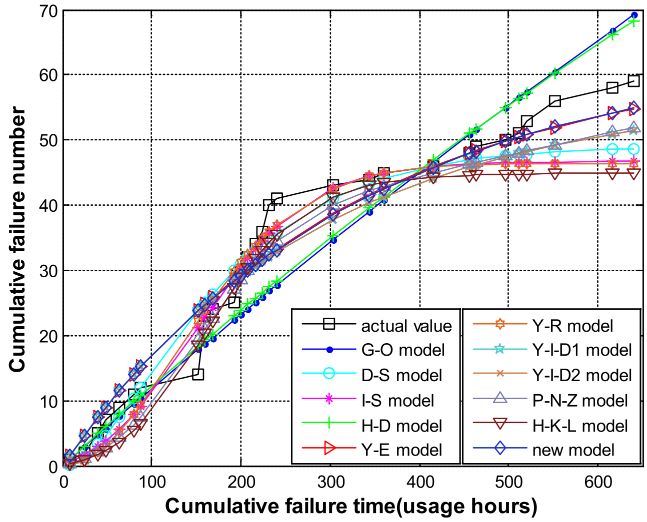

FIGURE 7. Fitting and predictive curves of several models.

Viewed from TABLE 10, SSE (fitting) together with R^2 index of new model is in middle level among that of all models. New model has more parameters than other model except H-K-L. It means new model's weighted comprehensive evaluation index (i.e., AIC or BIC) from fitting accuracy and the number of parameters is not good on this group of data. However, the SSE (prediction) together with predictive values show new model has the best fault predictive capability. Meanwhile, the curves shown in FIG. 7 also make intuitive description on performance of prediction.

The values of prediction can be divided into two types. The model G-O and H-D overestimate the development of fault trend while others underestimate the fault trend. With poor expressive capability of fault trend, the former exactly have relatively less parameters. Although the latter normally have good prediction capability on the fault development trend with relatively more parameters, the results have actually the infinitely varied difference. It seems that the model D-S, I-S, Y-R and H-K-L fall into over fitting to make their prediction early convergence, especially, the model H-K-L, having same number of parameters as new model, almost keeps unchanged in the final stage of prediction. As analyzed before, these four s-shape models normally perform well on s-shape fault data in aspect of fitting. However, they perform not so well in aspect of prediction. Viewed from scatter diagram, the fault data set has s-shape feature in early and middle stage while has apparent growth trend in late stage. Therefore, the early convergence of above s-shaped models can make explanations of the reason why their prediction capability are

TABLE 10. Fitting and predictive values obtained from several models based on the first 21 samples in fault data set [30].

Cumulative time (usage hours)	Cumulative faults	G-O model	D-S model	I-S model	H-D model	Y-E model	Y-R model	Y-I-D1 model	Y-I-D2 model	P-N-Z model	H-K-L model	new model
8	1	0.9605	0.1702	0.4896	1.0044	1.5676	0.0867	1.5983	1.5983	0.3287	0.2472	1.5786
24	2	2.8737	1.3690	1.6398	3.0006	4.5842	0.7737	4.6627	4.6627	1.1434	0.8765	4.6050
40	5	4.7766	3.4036	3.0481	4.9800	7.4499	2.1142	7.5589	7.5589	2.2114	1.7367	7.4700
48	7	5.7242	4.6398	3.8596	5.9634	8.8286	3.0105	8.9468	8.9468	2.8575	2.2755	8.8453
64	9	7.6118	7.4020	5.7148	7.9179	11.4827	5.2031	11.6077	11.6077	4.4063	3.6182	11.4880
80	11	9.4891	10.3970	7.8931	9.8559	14.0049	7.8453	14.1225	14.1225	6.3318	5.3741	13.9943
88	12	10.4240	11.9358	9.1042	10.8187	15.2187	9.3005	15.3276	15.3276	7.4433	6.4246	15.1991
152	14	17.8126	23.7680	21.0836	18.3772	23.8993	22.2954	23.8360	23.8361	19.2614	18.6112	23.8009
160	23	18.7250	25.0904	22.7225	19.3041	24.8665	23.8985	24.7707	24.7708	20.9127	20.4069	24.7590
168	24	19.6349	26.3665	24.3505	20.2272	25.8098	25.4623	25.6794	25.6794	22.5453	22.1966	25.6935
192	25	22.3499	29.9064	29.0398	22.9731	28.5019	29.8497	28.2564	28.2564	27.1730	27.3326	28.3621
200	30	23.2500	30.9890	30.5004	23.8807	29.3554	31.1931	29.0680	29.0681	28.5846	28.9133	29.0788
208	32	24.1477	32.0234	31.8946	24.7845	30.1878	32.4709	29.8570	29.8571	29.9173	30.4090	29.2088
216	34	25.0430	33.0101	33.2160	25.6844	30.9997	33.6807	30.6240	30.6241	31.1670	31.8126	30.0350
224	36	25.9358	33.9500	34.4598	26.5806	31.7916	34.8212	31.3697	31.3698	32.3324	33.1198	31.6281
232	40	26.8262	34.8441	35.6233	27.4730	32.5641	35.8917	32.0947	32.0948	33.4139	34.3288	32.3961
240	41	27.7143	35.6936	36.7051	28.3616	33.3177	36.8924	32.7994	32.7995	34.4139	35.4401	33.1458
304	43	34.7327	41.0529	42.6457	35.3372	38.7185	42.5699	37.7751	37.7752	40.0450	41.2784	38.5369
344	44	39.0427	43.3328	44.4998	39.5787	41.5906	44.4266	40.3601	40.3601	42.1949	42.9853	41.4212
360	45	40.7504	44.0650	44.9874	41.2504	42.6441	44.9145	41.2961	41.2962	42.9005	43.4215	42.4830
416	46	46.6552	46.0000	46.0000	46.9914	45.9498	45.8840	44.1856	44.1857	45.0024	44.3042	45.8314
456	48	50.8058	46.9258	46.3357	50.9894	47.9875	46.1604	45.9266	45.9266	46.3208	44.5867	47.9103
464	49	51.6283	47.0763	46.3808	51.7789	48.3661	46.1931	46.2462	46.2463	46.5752	44.6240	48.2980
496	50	54.8991	47.5846	46.5130	54.9093	49.7913	46.2785	47.4383	47.4383	47.5745	44.7322	49.7622
512	51	56.5212	47.7905	46.5576	56.4467	50.4535	46.3025	47.9856	47.9857	48.0663	44.7681	50.4452
520	53	57.3290	47.8832	46.5759	57.2132	50.7727	46.3115	48.2480	48.2480	48.3109	44.7828	50.7752
552	56	60.5385	48.1949	46.6296	60.2471	51.9748	46.3335	49.2262	49.2262	49.2825	44.8253	52.0216
616	58	66.8541	48.6053	46.6806	66.1633	54.0558	46.3467	50.8805	50.8804	51.2089	44.8648	54.1969
640	59	69.1874	48.7074	46.6897	68.3308	54.7380	46.3479	51.4106	51.4106	51.9285	44.8716	54.9156
AIC		67.5725	58.3595	53.6460	68.9233	64.6544	59.2544	64.7108	64.7107	58.4994	62.6367	70.8873
BIC		69.6616	60.4485	56.7796	72.0568	67.7879	62.388	67.8444	67.8443	62.6775	68.9035	77.1544
SSE (fitting)		818.233	207.597	150.817	714.458	333.330	190.792	384.316	384.312	260.621	252.406	341.360
SSE (prediction)		290.778	302.288	450.271	259.867	55.6289	481.232	204.283	204.283	186.427	661.929	52.788
R^2		0.8316	0.9573	0.9690	0.9246	0.9314	0.9607	0.9209	0.9209	0.9464	0.9481	0.9297

not so good on this data set. Accordingly, new model together with model Y-E has concave feature which performs not so well in aspect of fitting while performs better in aspect of trend description on this s-shape data set.

The slowly increasing trend together with relatively small value of reliability curves shown in FIG. 8 (a) indicates the reliability of this software is not good on stage of collecting fault date set. Viewed from FIG.8 (b), new model's reliability evaluation value is in the middle level, similarly matching the values and trends to Y-E, Y-I-D1 and Y-I-D2. G-O and H-D show relatively small values and slowly increasing trend of reliability. At the other extreme, D-S, I-S, Y-R and H-K-L show abnormal characteristic, i.e., the reliability value increases to 1 dramatically in middle testing stage while almost keeps unchanged in late testing stage. Similarly, model H-K-L, P-N-Z, Y-R, I-S and D-S show exceptions—declines of reliability on stage of early testing phase.

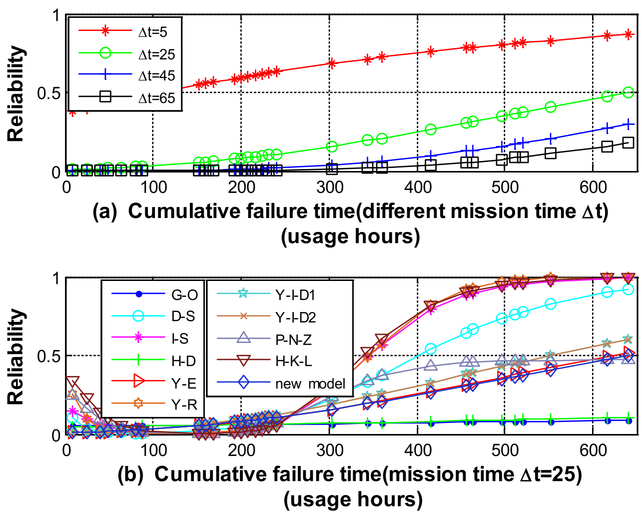

FIGURE 8. Reliability curves based on proposed model (a), and on several models under mission time Δt=25 usage hours (b).

5) FAILURE DATA FROM SOURCEMONITOR SOFTWARE (CASE 5)

In this section, we select an updating failure data set from a freeware named as SourceMonitor to show the effectiveness of new model. Several versions of the software have been updated since the first version was released in 2000. There were totally 276 faults shown on a sharing platform provided by Campwood Software till on June 5, 2020 [31]. We select recent 134 fault data (i.e., from the fault time on

TABLE 11. Pre-processed failure data based on selected fault data from sourcemonitor software [31].

Fault number	TBF (days))	CTBF (days)	Fault number	TBF (day(s))	CTBF (days)	Fault number	TBF (day(s))	CTBF (days)
2	56	56	52	42	643	93	13	1930
3	6	62	53	20	663	94	55	1985
4	77	139	54	1	664	95	118	2103
5	11	150	55	7	671	96	32	2135
6	7	157	56	21	692	97	11	2146
7	8	165	57	25	717	98	23	2169
9	2	167	58	5	722	99	50	2219
10	20	187	59	12	734	100	3	2222
11	5	192	60	52	786	101	1	2223
13	6	198	61	11	797	102	2	2225
14	2	200	64	6	803	103	98	2323
15	2	202	65	1	804	104	82	2405
20	4	206	66	7	811	105	20	2425
21	1	207	68	6	817	106	57	2482
23	10	217	69	4	821	107	1	2483
24	1	218	70	49	870	108	18	2501
26	7	225	71	25	895	109	37	2538
27	2	227	72	6	901	110	29	2567
30	18	245	73	6	907	111	23	2590
31	6	251	74	9	916	114	3	2593
32	3	254	75	35	951	115	45	2638
33	3	257	76	18	969	116	61	2699
34	2	259	77	16	985	117	12	2711
35	20	279	78	19	1004	118	43	2754
36	2	281	79	25	1029	119	50	2804
37	1	282	80	18	1047	120	7	2811
38	36	318	81	82	1129	122	70	2881
39	2	320	82	7	1136	123	16	2897
42	25	345	83	18	1154	124	164	3061
43	4	349	84	108	1262	126	186	3247
44	62	411	85	21	1283	127	4	3251
45	34	445	86	61	1344	128	3	3254
46	9	454	87	60	1404	129	24	3278
47	2	456	88	366	1770	130	1	3279
48	31	487	89	10	1780	131	27	3306
49	97	584	91	95	1875	133	16	3322
51	17	601	92	42	1917	134	30	3352

May 28, 2011 to the fault time on June 5, 2020), listed and explained on platform, as research subjects. For convenience to pre-process these fault data, we select the unlisted fault time found on Apr. 2, 2011, which is close to the first data of this failure data set, as a reference point. The pre-processed fault data are shown in TABLE 11(note: the total fault number is combined into 111 for the fact that some same faults occurred on the same day).

Considering the 70^{th} fault data which corresponds to the CTBF 1404 is far away the 71^{th} fault data which corresponds to the CTBF 1770, so, we select the first 70 fault data in TABLE 11 to fit parameters, then, the last 41 data are used to assess models' predictive power. The estimated parameters of several existing models are shown in TABLE 12, and corresponding fitting and prediction capabilities with AIC, SSE and R^2 are listed in TABLE 13. Several fitting and prediction curves are shown in FIG.9, and reliability curves are shown in FIG. 10 based on MTBF (i.e., 20.3478).

Observed from TABLE 12, the initial error number achieved from new model is larger than found fault number while that from others is close to or even smaller than found number. According to current development trend of finding fault data, it is a high probability event that the initial error number is far larger than found fault number.

Compared with previous fault data sets, current fault data set is relatively complex: (i) Current fault data has multi-stage-growing feature. Furthermore, the first part(for fitting) of current data set is with a concave trend although it is with a bit of s-shape feature, so, having concave characteristic, new model together with G-O, H-D and Y-E is apparently superior to other models in aspect of fitting. (ii) All models perform not so well in aspect of prediction. The essential reason consists in too many data (for prediction) with variant growing-rate and models' inherent deficiency—only for short prediction.

The SSE results listed in TABLE 13 exactly show new model has best fitting and prediction capabilities although the SSE (fitting) value of new model is almost equal to that of G-O, H-D and Y-E model, and this superiority is also shown in FIG. 9.

TABLE 12. Mle solutions of several existing models based on the first 70 samples in TABLE 11.

Model	Parameters $\hat{a}_0$	$\hat{p}$	$\hat{q}$	$\hat{b}$	$\hat{\alpha}$	$\hat{\beta}$
G-O	109.08	-	-	1.1380×10^{-3}	-	-
D-S	91.155	-	-	3.4584×10^{-3}	-	-
I-S	93.325	-	-	2.5279×10^{-3}	-	1.5798
H-D	109.03	-	-	1.1388×10^{-3}	9.2216×10^{-5}	-
Y-E	109.17	-	-	202.19	-	5.6280×10^{-6}
Y-R	211.65	-	-	0.5561	-	4.2425×10^{-6}
Y-I-D1	105.67	-	-	1.1787×10^{-3}	5.9924×10^{-7}	-
Y-I-D2	105.75	-	-	1.1777×10^{-3}	1.1103×10^{-5}	-
P-N-Z	93.590	-	-	2.5203×10^{-3}	4.7279×10^{-8}	1.5697
H-K-L	1228.8	1.2807	0.1399	5.2476×10^{-2}	3.4935×10^{-3}	414.10
new model	53149	6.7230×10^{-2}	0.6258	2.3361×10^{-6}	1.1377×10^{-3}	1.1775×10^{-5}

Note: The product expression $r\alpha$ in model Y-E and Y-R is replaced with b, the parameter c in model H-D is replaced with α, and the parameter W and A in model H-K-L are replaced with p and q respectively.

TABLE 13. Comparison of goodness-of-fit and predictive capability.

Model	AIC	BIC	SSE (fitting)	SSE (prediction)	R^2
G-O	247.94	252.44	1882.7	6941.8	0.9577
D-S	242.77	247.27	3002.9	22681.2	0.9326
I-S	247.78	254.52	1932.0	19621.3	0.9566
H-D	249.94	256.69	1882.9	6950.8	0.9577
Y-E	249.95	256.70	1885.8	6931.0	0.9576
Y-R	256.98	263.73	6424.4	23940.2	0.8557
Y-I-D1	249.98	256.73	1963.5	9044.7	0.9559
Y-I-D2	249.99	256.73	1962.9	9079.0	0.9559
P-N-Z	249.78	258.77	1930.6	19231.0	0.9566
H-K-L	258.49	271.98	4330.8	30403.1	0.9027
new model	255.94	269.43	1882.6	6929.4	0.9577

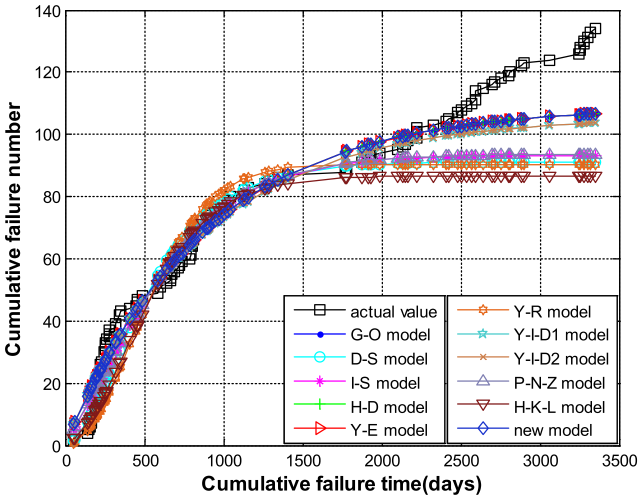

FIGURE 9. Fitting and predictive curves of several existing models.

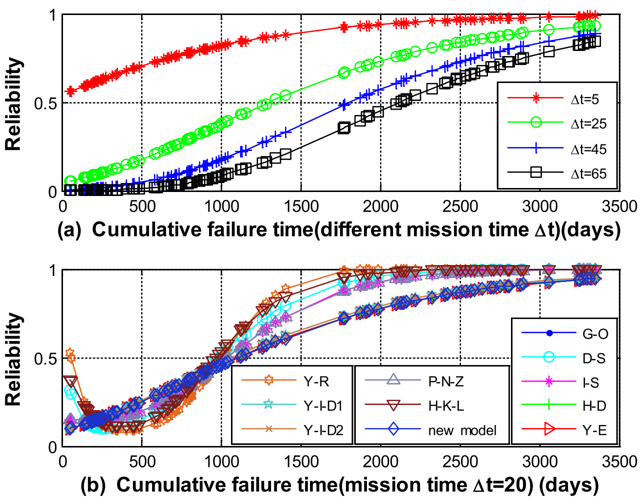

FIGURE 10. Reliability curves based on proposed model (a), and on several models under mission time Δt=20 days (b).

In FIG. 10(b), the reliability curves of new model, G-O, H-D, Y-E, Y-I-D1 and Y-I-D2 show steady-state growth trend which reveals certain rationality of reliability growth. As for assessing quality, similar to previous cases, some models such as I-S, D-S, Y-R, P-N-Z, and H-K-L perform with abnormal characteristics, i.e., high reliability in late testing stage and decline in early testing stage.

C. MODEL EVALUATION

In this section, we give several comments from effectiveness of fitting and prediction, capability, applicability, quality of assumptions and simplicity to describe the characteristics of new model together with other models.

1) EFFECTIVENESS OF FITTING AND PREDICTION

As illustrated by making comparisons with existing models, It seems that the fitting capability of new model on case 2, 3 and 4 is not satisfying, but the prediction capability on these cases is the best; Except PNZ model, new model has the better fitting capability than other models, and has the best prediction capability on case 1; As for case 5, new model has the best fitting and prediction capability.

For fitting and/or prediction effectiveness, we attempt to clarify some obvious bugs in existing literature. The SSE (prediction) from Z-T-P model on case 1 is 495.98 [18]. However, this result is questionable because the real SSE should be 686.82 due to the fact that the last two data are larger than the corresponding results from P-N-Z model [12] while the other values from those two models are almost the same. In addition, some prediction values from G-O model on case 1 [12] are also questionable. After all, G-O model is an exponential growth model, and the phenomenon that prediction values increases and then decreases (for example, the partial prediction values are 129, 134 139 138 135 133) should not take place.

2) CAPABILITY

Excessive pursuit of model's fitting ability perhaps is not necessarily a good thing because generalization ability of over-fitting is normally unsatisfactory. The SSE values of model I-S and P-N-Z on case 2 and 4 and model D-S on case 3 meet such conclusion well.

As a matter of fact, new model, which has 6 parameters, is more complex than some existing models. Viewed from AIC (BIC) criterion, the degree of freedom is taken into consideration by assigning model with more parameters a larger penalty. Therefore, the more parameters, the larger value of AIC (BIC) may be to a great extent. It seems that multi-parameter model likely brings certain negative effect to its likelihood capability. This is just the reason why new model seems to be less fitting than some simple models on certain data-sets based on AIC (BIC) criterion. However, multi-parameter model normally has better descriptive power than other less-parameter model on fault development trend. Generally speaking, a complex model, especially incorporated with imperfect debugging, fault detection rate per error, error introduction rate, and so on, is probably effective because it better explains actual running situation of software, therefore, it generally has better capability of short-term prediction than that of traditional models [32]. In this sense, new model, a complex failure MVF with multiple parameters, has robust descriptive capability in aspect of fault prediction.

Strictly speaking, model P-N-Z, D-S, Y-R and H-K-L are not SRGMs in a certain extent, at least in early stages of software failures. This judgment can be verified from the results on case 1, 3, 4 and 5. Further research reveals the reliability predictions of model P-N-Z and D-S on case 2 also show short-term decline with time in early testing and modification stage. In this way, new model, keeping steady-state reliability growth trend, is a SRGM with good capability in reliability growth.

In addition, compared with some simple model with less-parameters, new model, if applied in a software product, may cost more solving time while can get better failure development trend of software. More solving time almost does not lead to more human costs while more effective prediction trend of software failure can advance release time. Viewed from this point, new model has positive impact of amortized cost if it is approached practically.

On the whole, compared with some simple models, new model generally has better descriptive capability in aspect of fault prediction, reliability growth and positive impact of amortized cost.

3) APPLICATION

Constrained by the complex factors such as construction, function, scale, developing and running environments, testing, collection method of fault data and so on, Actual fault data sets are always with many different characteristics, and there rarely exists same fault behavior and procedures. Therefore, the essential significance of software reliability modeling is to find a model to adapt the fault (failure) development trend of different software product.

According to existing research result, there are two typical fault classes [25], i.e., concave and s-shaped models. In addition, LPETM, a less-common class of model, mainly focuses on infinite fault mode. Partial fault data (for fitting) from case 1 and case 5 can be roughly classified as concave models, and fault data (for fitting) from case 2, 3, and 4 as s-shaped models. Viewed from the results of fitting and prediction, model I-S, D-S, Y-R and H-K-L have s-shape characteristics while new model together with G-O, H-D, Y-E, Y-I-D1 and Y-I-D2 model has concave characteristic. So, the fitting capability of the former is normally superior to the latter on case 2, 3 and 4. Correspondingly, the fitting capability of the former is obviously inferior to the latter on case 1 and 5. It's worth noting that prediction capability of the former is obviously inferior to the latter on case 2, 3 and 4 due to the fact that prediction of the former normally falls into precocity to approximately keep constant. As for P-N-Z model, it normally performs well in fitting while not so good in prediction on many cases.

On the whole, new model has similar performance with model G-O, H-D, Y-E, Y-I-D1 and Y-I-D2. However, the superiority of new model consists in its best fitting and prediction capability on case 1 and 5 and excellent prediction capability on almost all cases. Such superiority means new model has better application than other models in aspect of certain-range prediction on certain software with different development and usage environments.

4) QUALITY OF ASSUMPTIONS

In process of modeling, we put forward several assumptions to form the basis of new model. Although those assumptions are normally different from traditionally opinions and are difficult to be verified, the viewpoints meet with some engineering experiences in certain degree. For example, the reason why the error modification efficiency and fault detection rate per error are normally assumed as exponential decay functions lies in such a phenomenon that the fault detection and error modification likely become more and more difficult. Although other reasons seem not always to be reasonable, such phenomena that the detection and modification efficiency decrease actually exist. In addition, the assumptions we put forward have explicit mathematical meaning and physical meaning as well. So, viewed from the engineering experiences, logical consistency and model verification on given cases, the reasonability of proposed assumptions should not be unthinkingly denied.

5) SIMPLICITY

According to the basic indices of simplicity, we make comment as follows: (i) Compared with traditional models, new model has no special requirements in fault data collection. (ii) In new model, detection and modification are fully considered by combining with fault detection rate per error, error introduction rate and error modification efficiency. Although new model seems to be abstract and complex, its concepts and parameters are still easy to be understood and explained. (iii) Compared with some existing models, new model with more parameters is in a relatively inferior position in solving accuracy and consuming time.

Of course, viewed from number of parameters, new model with 6 parameters may not be normally taken as simplicity. However, too simple model may easily mask some important factors related to fault (failure), and likely weaken its capability of fault (failure) explanation.

V. RELATED WORK

A. INTRODUCTION AND REVIEW OF EXISTING TYPICAL MODELS

Over the past decades, researches on SRGM have been greatly conducted and developed, and a large number of outstanding contributions on the theories and applications have been made. So far, SRGM is one of the most active and successful models in field of software reliability analysis, prediction, application and management. The reason why it is widely concerned is not only for its description and explanation of reliability information (such as the reliability at any time for given mission time, the number of residual error, the cumulative number of detected fault and error modification, etc.), but for its provision of important decision on cover degree of testing resources, cost control, decision of optimal release time [33], [34], [35].

There were two completely different mechanisms used for modeling SRGMs. In traditional viewpoints, fault rate is considered as a function of the number of residual errors, and all the fault rates per error are the same. On the contrary, another viewpoint, proposed by Littlewood [36], believed the failure rate was a random variable. Such viewpoint contains three typical assumptions: (i) The uncertainty of software reliability growth is the result of execution profile rather than the result of error modification. The error triggered with large probability will be modified, and error modification will lead to the decrease of fault rate; (ii) Each of faults (failures), considered to obey exponential distribution, will be solely triggered by an error; (iii) The fault rate of each error obeys Gamma distribution, then, the fault rate per error obeys Pareto distribution based on Bayesian theory. Thus, the derived fault rate of software decrease as time goes on.

Traditional SRGM, based on Markov theory, was originally derived from Musa's basic time execution model and G-O model. As a typical representative of existed exponential models, G-O model was applied by Musa to analyze the performance of real-time command and control system [4]. Considering the relationship between execution time and calendar time, Musa et al. [37] used testing compression factor divided by total fault number and MTTF to express fault detection rate per error in testing phase. Ohba [9] presented hyper-exponential growth model by summing MVF of modules which have different initial error number and fault rate. Yamada and Osaki [10] expended to suppose the fault intensity in different module is not the same, and the fault intensity in same module remains the same. A V-tub-shaped function proposed by Pham was taken as the fault detection rate [38]. Above models are based on NHPP framework whose total error number is regarded as a constant and the error modification is perfect.

Ohba and Yamada [5], Pham and Zhang [11], Pham et al. [12], and Zhang et al. [18] applied inflection S-shaped fault detection rate per error to their models. In these models, the fault detection rate per error is taken as a non-decreasing function due to so-called ''learning'' phenomena. As mentioned in Introduction, ''learning'' process seldom takes place in actual test environment. Furthermore, the phenomenon that finding faults is increasingly difficult during the testing phase still exists to certain extent.

Based on an assumption that new errors are likely introduced when debugging the detected faults, Yamada et al. [7] presented several imperfect debugging models to focus on the variability of total error number which was generally considered as an exponential or linear increasing function. Pham et al. [12], Pham and Nordmann [14] constructed a generalized imperfect debugging model to form a typical framework which mainly contains total error number function and fault detection rate function. On the basis of generalized imperfect debugging model, Zhang et al. [18] further considered error modification efficiency. In their models, the increasing rate of total error number and the error modification rate are all proportional to the increasing rate of found fault number.

On the basis of defining error reduction factor [20], which has the form of decreasing exponential function of detected error number, Musa and Okumoto [21] proposed an infinite fault model(i.e., LPETM). Although it is impossible that any software with finite codes has infinite errors, it is possible that imperfect debugging together with the introduction of new errors may lead to infinite faults in theory. Essentially speaking, error reduction factor was defined to illustrate the error modification efficiency.

With the development of software reliability research, research on SRGM was gradually considered to involve several factors such as Imperfect Debugging (ID), Testing-Effort (TE) and Change Point (CP), and so on. The influence of CP on SRGM is not only reflected in the description of testing environment (i.e., fault detection rate per error which has many forms of functions such as inflection S-shaped function, multiple CP function [39]), but in the description of testing-effort such as multiple CP of TE [40]. Besides logistic type and some extensive types of Testing-Effort Function (TEF) [15], [41], [42], some s-shaped TEF models were adopted to describe cumulated testing-consumption amount with a flexible varying trend [24], [43]. Huang et al. [44] introduced TE described with multiple CP and ID into SRGM, and the ID, described as the change rate of total error function, is proportional to the fault detection rate. In these models, the total error number is either regarded as a constant or as a function which is proportional to TEF or detected fault number. Considering the result of learning effect on testing resources and efforts, Huang et al [35] proposed an imperfect debugging SRGM by means of introducing fault detection rate per error with form of CP, cyclical error fluctuation rate and fault detection rate, which of last two physical quantities, taken as product factors, decide the change rate of total error number.

Establishing a relatively uniform framework of SRGM to cover multiple models is an interesting research field of software reliability fitting and prediction. Considering the impact of various factors, Zhang et al. [45] proposed a framework covering various SRGMs. Such unified framework describes a blueprint for modeling of software reliability. However, there is a long road to go because of too many uncertain factors which inevitably increase the complexity of interpretation, solution and analysis.

With the developing of software reliability research level, modeling SGRM gradually aims to explore the quantitative relationship between reliability and internal mechanism from fault detection to fault (error) removal. Roy et al. [46] combined s-shaped fault detection rate with improved testing learning process under imperfect debugging model. Pham [47] adopted loglog fault detection rate together with testing coverage for software reliability modeling. Lo [48] proposed a unified software reliability modeling method including fault detection and removal based on imperfect debugging. The general characteristic of these models is to concern the number of detected faults till found faults (error) are removed completely.

Many scholars focused on instability of test environment to take actual random factors into account. Li and Pham [49] adopted a random time-independent variable to make description of uncertain affection of fault detection rate. Considering the fluctuation and irregularity of intensity failure and collision level function, Yamada [50] made research of stochastic differential equation model with Wiener process to describe actual fault detection on open source solution, considering the fluctuation and irregularity of intensity failure and collision level function. Furthermore, under belief reliability theory, Liu et al. [51] proposed a software belief SRGM to deal with epistemic uncertainty based on uncertain differential equation, by considering the affection of irregular fluctuation and Liu Process [52] on testing processing. Differing from traditional viewpoint that fault detection rate per error is considered as a defined function under software testing and field operation environments, Teng and Pham [53] considered random effects on fault detection rate by means of introducing a random-distributed variable to cover both the testing phase and the operating phase in the software development cycle. Taking into account uncertainty of field environment and external input, Li et al. [54] established a reliability model described as Open Stochastic System (OSS), adopting a load of gamma distribution together with interference with white noise on fault detection rate and description of input with Gaussian distribution.

In general, service for fault detection or error modification is more or less delayed. Considering arrival of detected faults with a certain rate to enter waiting queue according to a certain priority, Lin and Huang [55] adopted single queue theory to model software debugging behavior of multi-channel system. Lin et al. [56] proposed a preemptive priority queuing model, considering the faults assigned higher priority would be able to preemptively acquire resources already occupied by lower priority faults. Furthermore, instead of adopting model-based approaches, Lin and Huang [57] proposed a queuing-based simulation strategy to investigate the fault correction process and provide system performance information based on staffing level, average response time and average waiting time.

Besides traditional numerical methods, evolutionary search methods such as genetic algorithm [58], swarm intelligence optimization algorithm [43], etc., appeared for making parameter estimation. By means of adopting affine-combination mutation and uniform crossover scheme, Yaghoobi et al. [59] proposed a modified differential evolution algorithm for making solutions of MLE. Combining the quick-converge advantage of PSO(Particle Swarm Optimization) with the advantage of high solving accuracy, good stability and strong robustness of SSA(Sparrow Search Algorithm), Yang et al. [60], proposed a hybrid SSA-PSO algorithm to improve the solution accuracy and convergence speed of parameter estimation by means of adopting two-stage strategy. However, no matter what type of optimization methods is adopted, making solutions of non-linear equation sets derived from MVF $m(t)$ always faces some challenges such as selection of initial value, application of technique to improve convergence speed and solution accuracy and adoption of strategy to get satisfactory global solutions, etc.

In most software reliability growth studies, pursuit of model optimization based on some typical criteria always is the fundamental motivation. For the purpose of finding optimal model and the best overall ranking, a hybrid approach (i.e., model selection strategy), named as CODAS-E (Entropy-Combinative Distance-Based Assessment), used for getting order preference by similarity to ideal solution and analytic hierarchy process, was proposed to select and rank SRGMs based on multiple performance indexes [61]. On the contrary, Taken as one of the representatives of early empirical software engineering research, a so-called u-plot technique together with prequential likelihood method, adopted to allow a user to decide upon the most appropriate model for each application, was presented to stress the accuracy of model predictions rather than attempting to decide which model is generally best [62]. Although modeling theory, method and technology determine the quality of model, the characteristics of collected data also affect the results of model. Applied as an empirical judgment strategy based on making statistical analysis to identify the NHPP feature of fault data-set, a two-phase method, within which goodness-of-fit to Poisson distribution, existence testing of serial dependency and stability testing base on Laplace trend checking are the important operation processes, was proposed to examine whether the failure data fits NHPP-based SRGM or not [63].

Viewed form implementation, selection of optimal model, model decision and characteristic analysis on data-set, which of them play an important role in software reliability engineering, are with practical significance and great development potential.

B. FEATURES AND TRENDS OF EXISTING RESEARCH

According to above analysis, some research directions with respect to SRGM can be briefly outlined as follows:

(i) Model structures of SRGMs evolved from simple forms to complex ones. Scholars tried to establish a unified framework of SRGM, the core of which consisted in making construction of a definitive relationship among the fault (failure) number, fault (failure) detection rate per error, total error number, modified error number, under the conditions of ID, TE and CP. This unified framework makes a description of behavior and impact factors on fault (failure) detection, modification and error introduction.

(ii) Taken as an important research direction with terminal goal, modeling of SRGM begin to focus on describing internal failure mechanism and process of reliability growth from fault detection to fault (error) removal.

(iii) Statistical distributions (including queuing theory), used for making description of the randomness of fault (failure) detection, error modification, uncertain impact of input environment (or execution profile), testing cost, software release, even management strategy, are comprehensively considered in modeling of SRGM.

(iv) Intelligent optimization methods, taken as a sort of solution strategies by means of searching population and individual iteration optimization, has been applied to parameter estimation so as to improve the solving efficiency.

(v) Selection of optimal model or decision together with characteristic analysis on fault data-set based on special method, strategy and technology is a noteworthy field.

VI. SUMMARY AND CONCLUSION

A. MOLDING THREAD OF NEW MODEL

In this paper, being lighten by the first two research directions generalized above and taking NHPP generalized imperfect debugging framework as a basis, we carry out a serial of researches for modeling a failure MVF, forming the core of novel SRGM.

Viewed from research process, the main threads of our work are shown as follows:

NHPP framework → basic (or public) assumption (i.e., Eq.(8)) → new assumptions(i.e., Eq.(9),(10) and (11)) → new model(i.e., Eq.(12)) → failure MVF $m(t)$(proof and discussion) → parameter estimation(from public fault data sets) → test and verification(fitting, predicting and getting reliability) → analysis, comparisons and conclusions.

B. FEATURES OF NEW MODEL

In respect to modeling methods, we mainly focus on making of new assumptions, explanations and formation of quantitative constraint relations to form new failure MVF based on NHPP generalized imperfect debugging framework. In new model, the viewpoints, in which the fault detection rate per error, total error number, modified error number and their corresponding constraint relations are normally different from existing viewpoints, are shown as follows:

(i) Differing from existing models whose error modification probabilities are generally considered as a constant between 0 and 1, we take error reduction factor as error modification efficiency to denote the relationship between detected fault number and modified error number.

(ii) Making full consideration of the dynamic process which contains new error introduction of and error modification, we take $[a(t) - x(t)]'$ rather than $x'(t)$ to describe the error modification intensity. On the basis of the fact that the error modification process may be accompanied with introduction of new errors, and of the positive correlation between error modification and error introduction, we presented a new viewpoint that the total error introduction rate is proportional to the change rate of generalized residual errors to make description of error introduction. In existing models, either the total error function is a deterministic function of time or the change rate of total error is only proportional to the change rate of modified errors (i.e., $x'(t)$). This is the fundamental difference between new model and traditional models.

(iii) Differing from existing models whose fault detection rate per error is either a non-decreasing function (such as inflection s-shape function of time) or CP function, we adopt exponential decay function of time as fault detection rate per error based on making consideration of existing phenomena such as seldom ''learning'' of detection, difficulty in finding residual errors and reliability degradation of testers themselves.

C. CONLCUSION

According to novel modeling viewpoints, method, structural analysis, parameter interpretation, performance verification and comparison in context of this research, several conclusions can be drawn as follows:

(i) Different from traditional generalized imperfect debugging NHPP model and LPETM, new model can be considered as a generalized LPETM which has infinite fault characteristics while can converge to an upper bound.

(ii) The derived failure MVF not only shows robust ability of fitting fault data to a certain range and degree, but also performs better than some existing models on short-term fault prediction on certain data sets.

(iii) New model is a relatively effective SRGM. Prediction on each case normally meets with actual running trend of software.

In general, based on above modeling contents, the SRGM we derived enhances the diversity of existing models.

VII. FURTHER WORK

In fact, it is impossible that software reliability model under assumptions always meets with actual running mode. In most cases, operational processes (such as fault detection, error modification and error introduction, etc.) on software, has obvious characteristics of stochastic process. In our research, the proportionality coefficient (i.e., $-p$) between the error introduction rate and the change rate of generalized residual errors is lack of flexible expression. Inspired by the ideas of preemptive priority queuing model [56], queuing-based simulation strategy [57] in aspect of error modification and learning effect on testing resources and efforts for total error introduction rate decided by instantaneous error fluctuation rate and fault detection rate per error [35], we will make supplement for current work to explore more rational error introduction rate per modified error.

It is meaningful to check how the errors (defects) are introduced and/or fixed (modified), whether the ''learning effect'' takes place or not, and how this process impact the debugging time, debuggers' skills, testing, even debugger, the ''differences'' among faults in terms of easiness of being detected and debugged, the non-linear relation between testing effort and testing time, and so on. This work, with great challenge and relative dependence, will be done in our next research phase.

Moreover, the parameters obtained from MLE are only called as point estimation parameters. People usually expect to get interval estimation through which the truth-value will be in certain interval with given confidence. Under the condition of enough samples, MLE has a feature of asymptotic normality (i.e., the estimated parameter approximately obeys normal distribution) which can be used for interval estimation. In general, the interval estimation of multiple parameters can be indirectly obtained from calculating Fisher information matrix [64]. However, for small or moderate samples, it's not suitable for directly applying such method to interval estimation because these samples seldom meet the condition of normal distribution.

We will go on to research interval estimation of parameter to increase the integrality of our current research.

To achieve this goal, making variable transformation or selecting approximation method based on likelihood ration may be a reasonable method. There are several problems should be solved well: (i) How to distinguish and decide the scale of samples (i.e., How many samples can be regarded as small, moderate or large samples)? (ii) Under the small or moderate samples, how to find proper variables to transfer estimated parameters into new statistics which approximately obey normal distribution? As an approximate alternative solution, likelihood ration, defined as $-2\ln(\theta_k \big/ \hat{\theta}_k)(k = 0, 1, \cdots, n)$, obeys chi-square distribution with $n+1$ degrees of freedom. (iii) For any parameter $\hat{\theta}_k$, how to get its $100(1-\alpha)$ percent confidence interval? All these problems, involved in lots of theoretical and practical research from mathematical statistics, may be meaningful.

In addition, viewed from the fitting and predictive results, there are two cases, i.e., either underestimation or overestimation. It naturally points to a decision problem. Only according to static evaluation criteria such as AIC (BIC), SSE, R^2, and so on, is the selected model really convincing? Furthermore, is selected data-set really suitable for modeling SRGM based on NHPP framework? Inspired by the u-plot approach [62], best overall ranking to get optimal model [61] together with data type judgment [63], we plan to further combine more software engineering experiences with data analysis, model selection, accuracy concern and applicability judgment to make decisions that fit actual situation in modeling of software reliability.

APPENDIX
DERIVATION FOR FAILURE MVF

According to the discussion given in Section III, the proof process is based on following typical constraint relationships and initial conditions.

$$\begin{cases} m'(t) = b(t)[a(t) - x(t)] & < i > \\ \dfrac{x'(t)}{m'(t)} = qe^{-\beta m(t)} & < ii > \\ \dfrac{a'(t)}{[a(t)-x(t)]'} = -p & < iii > \\ b(t) = be^{-\alpha t} & < iv > . \end{cases}$$
$$m(0) = 0; x(0) = 0; a(0) = a_0 \tag{A-1}$$

The 3th expression in Eq. set (A-1) can be rewritten as following form of equivalent form:

$$a'(t) - x'(t) = (p-1)x'(t). \tag{A-2}$$

By substituting the 2nd expression in Eq. set (A-1) into right side of Eq. (A-2), a differential equation can be gotten as follow:

$$a'(t) - x'(t) = (p-1)qe^{-\beta m(t)}m'(t). \tag{A-3}$$

Through integrating two sides of Eq. (A-3) from 0 to t with initial conditions $m(0) = 0$, $x(0) = 0$ and $a(0) = a_0$, the solution of Eq. (A-3) can be obtained as follow:

$$a(t) - x(t) = a_0 - \frac{q(1-p)}{\beta}[1 - e^{-\beta m(t)}]. \tag{A-4}$$

By substituting Eq. (A-4) into the 1st expression of Eq. set (A-1), a differential equation about MVF $m(t)$ is obtained as follow:

$$m'(t) = b(t)\left\{a_0 - \frac{q(1-p)}{\beta}[1 - e^{-\beta m(t)}]\right\}. \tag{A-5}$$

By using separation of variables, Eq. (A-5) can be solved through sequential steps shown as follow:

$$\frac{d[m(t)]}{a_0 - \frac{q(1-p)}{\beta}[1 - e^{-\beta m(t)}]} = b(t)dt. \tag{A-6}$$

Operating integral substitution method on Eq. (A-6), we can further transform it into following form:

$$\frac{d[e^{\beta m(t)}]}{q(1-p) + [\beta a_0 - q(1-p)]\, e^{\beta m(t)}} = b(t)dt. \tag{A-7}$$

Suppose $k = q(1-p)$, $z = \beta a_0 - q(1-p)$, Eq. (A-7) can be transformed into a simple form shown as follow:

$$\frac{d[k + ze^{\beta m(t)}]}{k + ze^{\beta m(t)}} = zb(t)dt. \tag{A-8}$$

Integrate both sides of Eq. (A-8) from 0 to t, we can get a result shown as follow:

$$\ln\left[\frac{k + ze^{\beta m(t)}}{\beta a_0}\right] = z\int_0^t b(\tau)d\tau. \tag{A-9}$$

Transform and simplify Eq. (A-9), a result can be gotten as follow:

$$m(t) = \frac{1}{\beta}\ln\left[\frac{\beta a_0 e^{z\int_0^t b(\tau)d\tau} - k}{z}\right]. \tag{A-10}$$

Substitute k and z for corresponding expression $q(1-p)$ and $\beta a_0 - q(1-p)$ into Eq. (A-10) respectively, we can get an expression as follow:

$$m(t) = \frac{1}{\beta}\ln\left[\frac{\beta a_0 e^{[\beta a_0 - q(1-p)]\int_0^t b(\tau)d\tau} - q(1-p)}{\beta a_0 - q(1-p)}\right]. \tag{A-11}$$

Substitute $b(t)$ for expression $be^{-\alpha t}$ and calculate integration, the final result of Eq. (A-11) can be derived as follow:

$$m(t) = \frac{1}{\beta}\ln\left[\frac{\beta a_0 e^{[\beta a_0 - q(1-p)]\cdot\frac{b}{\alpha}(1-e^{-\alpha t})} - q(1-p)}{\beta a_0 - q(1-p)}\right]. \tag{A-12}$$

$\square$

ACKNOWLEDGMENT

The authors would like to highly appreciate Editor-in-Chief, Associate Editor, and all anonymous reviewers whose comments and suggestions will greatly promote us to improve the quality of this article.

REFERENCES

[1] J. D. Musa, "A theory of software reliability and its application," *IEEE Trans. Softw. Eng.*, vol. SE-1, no. 3, pp. 312–327, Sep. 1975, doi: 10.1109/TSE.1975.6312856.
[2] 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, doi: 10.1109/TR.1979.5220566.
[3] A. L. Goel, "A summary of the discussion on 'an analysis of competing software reliability models,"' *IEEE Trans. Softw. Eng.*, vol. SE-6, no. 5, pp. 501–502, Sep. 1980, doi: 10.1109/TSE.1980.230791.
[4] H. Pham, *System Software Reliability*. London, U.K.: Springer-Verlag, 2006, pp. 183–188, doi: 10.1007/1-84628-295-0.
[5] M. Ohba and S. Yamada, "S-shaped software reliability growth models," in *Proc. 4th Int. Conf. Rel. Maintainability*, 1984, pp. 430–436.
[6] 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, doi: 10.1109/TR.1983.5221735.
[7] S. Yamada, M. Ohba, and S. Osaki, "S-shaped software reliability growth models and their applications," *IEEE Trans. Rel.*, vol. R-33, no. 4, pp. 289–292, Oct. 1984, doi: 10.1109/TR.1984.5221826.
[8] H. Xizhi, "The hypergeometric distribution model for predicting the reliability of software," *Microelectron. Rel.*, vol. 24, no. 1, pp. 11–20, 1984, doi: 10.1016/0026-2714(84)90631-0.
[9] M. Ohba, "Software reliability analysis models," *IBM J. Res. Develop.*, vol. 28, no. 4, pp. 428–443, Jul. 1984, doi: 10.1147/rd.284.0428.
[10] S. Yamada and S. Osaki, "Software reliability growth modeling: models and applications," *IEEE Trans. Softw. Eng.*, vol. SE-11, no. 12, pp. 1431–1437, Dec. 1985, doi: 10.1109/TSE.1985.232179.
[11] H. Pham and X. Zhang, "An NHPP software reliability model and its comparison," *Int. J. Rel., Quality Saf. Eng.*, vol. 4, no. 3, pp. 269–282, Sep. 1997, doi: 10.1142/S0218539397000199.
[12] 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, Jun. 1999, doi: 10.1109/24.784276.
[13] H. Pham and C. Deng, "Predictive-ratio risk criterion for selecting software reliability models," in *Proc. 9th Int. Conf. Rel. Quality Design*, 2003, pp. 17–21.
[14] H. Pham and L. Nordmann, "A generalized NHPP software reliability model," in *Proc. 3rd ISSAT Int. Conf. Rel. Quality Design*. Anaheim, CA, USA: ISSAT Press, 1997, pp. 116–120.
[15] 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. 2007, doi: 10.1109/TR.2007.895301.
[16] Y. Nakagawa, "A connective exponential software reliability growth model based on analysis of software reliability growth curves," *IEICE Trans.*, vol. J77-D, no. 6, pp. 433–442, 1994.
[17] H. Pham, *System Software Reliability*. London, U.K.: Springer-Verlag, 2006, pp. 194–195, doi: 10.1007/1-84628-295-0.
[18] X. 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, doi: 10.1109/TSMCA.2003.812597.
[19] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*. New York, NY, USA: McGraw-Hill, 1987, pp. 254–255.
[20] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*. New York, NY, USA: McGraw-Hill, 1987, pp. 298–301.
[21] J. D. Musa and K. Okumoto, "A logarithmic Poisson execution time model for software reliability measurement," in *Proc. 7th Int. Conf. Softw. Eng.*, Orlando, FL, USA, 1984, pp. 230–238. [Online]. Available: http://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ac955b5e6bdba0b664d9f0b99864ea9ae24fd3a1
[22] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974, doi: 10.1109/TAC.1974.1100705.
[23] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, Mar. 1978.
[24] 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, doi: 10.1109/JSEE.2015.00024.
[25] A. Wood, "Predicting software reliability," *Computer*, vol. 29, no. 11, pp. 69–77, Nov. 1996, doi: 10.1109/2.544240.
[26] M. Ohba, "Inflexion S-shaped software reliability growth models," in *Stochastic Models in Reliability Theory*. Merlin, OR, USA: Springer-Verlag, 1984, pp. 144–162.
[27] 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, doi: 10.1109/24.273589.
[28] 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, doi: 10.1080/00207729208949452.
[29] W. Ehrlich, B. Prasanna, J. Stampfel, and J. Wu, "Determining the cost of a stop-test decision (software reliability)," *IEEE Softw.*, vol. 10, no. 2, pp. 33–42, Mar. 1993, doi: 10.1109/52.199726.
[30] *IEEE Recommended Practice on Software Reliability*, IEEE Standard 1633^{TM}-2016, Standards Committee of the IEEE Reliability Society, 2017, p. 122, doi: 10.1109/IEEESTD.2017.7827907.
[31] *Campwood Software*. Accessed: Jun. 5, 2020. [Online]. Available: https://www.campwoodsw.com/smbugs.html
[32] H. Pham, *System Software Reliability*. London, U.K.: Springer-Verlag, 2006, p. 207, doi: 10.1007/1-84628-295-0.
[33] P. K. Kapur, O. Singh, and A. K. Shrivastava, "Optimal price and testing time of a software under warranty and two types of imperfect debugging," *Int. J. Syst. Assurance Eng. Manage.*, vol. 5, no. 2, pp. 120–126, Jun. 2014, doi: 10.1007/s13198-014-0221-x.
[34] T. Manjula, M. Jain, and T. R. Gulati, "Optimal release policies of delayed discrete reliability growth model with imperfect debugging," *J. Test. Eval.*, vol. 44, no. 3, pp. 1376–1382, 2015, doi: 10.1520/JTE20150043.
[35] Y.-S. Huang, K.-C. Chiu, and W.-M. Chen, "A software reliability growth model for imperfect debugging," *J. Syst. Softw.*, vol. 188, Jun. 2022, Art. no. 111267, doi: 10.1016/j.jss.2022.111267.
[36] B. Littlewood, "Stochastic reliability-growth: A model for fault-removal in computer-programs and hardware-designs," *IEEE Trans. Rel.*, vol. R-30, no. 4, pp. 313–320, Oct. 1981, doi: 10.1109/TR.1981.5221099.
[37] J. D. Musa and K. Okumoto, "Application of basic and logarithmic Poisson execution time models in software reliability measurement," in *Software Reliability Modelling and Identification* (Lecture Notes in Computer Science), vol. 341, S. Bittanti, Ed. Berlin, Germany: Springer, 1988, pp. 68–100. [Online]. Available: https://link.springer.com/content/pdf/10.1007/BFb0034287.pdf?pdf=core, doi: 10.1007/978-3-642-82846-1_9.
[38] 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, doi: 10.1080/02331934.2013.854787.
[39] C. Huang and M. R. Lyu, "Estimation and analysis of some generalized multiple change-point software reliability models," *IEEE Trans. Rel.*, vol. 60, no. 2, pp. 498–514, Jun. 2011, doi: 10.1109/TR.2011.2134350.
[40] C.-T. Lin and C.-Y. Huang, "Enhancing and measuring the predictive capabilities of testing-effort dependent software reliability models," *J. Syst. Softw.*, vol. 81, no. 6, pp. 1025–1038, Jun. 2008, doi: 10.1016/j.jss.2007.10.002.
[41] S.-Y. Kuo, C.-Y. Huang, and M. R. Lyu, "Framework for modeling software reliability, using various testing-efforts and fault-detection rates," *IEEE Trans. Rel.*, vol. 50, no. 3, pp. 310–320, Sep. 2001, doi: 10.1109/24.974129.
[42] C. Zhang, G. Cui, F.-C. Meng, H.-W. Liu, and S.-X. Wu, "A study of optimal release policy for SRGM with imperfect debugging," *J. Eng. Sci. Technol. Rev.*, vol. 6, no. 3, pp. 111–118, 2013.
[43] C. Jin and S.-W. Jin, "Parameter optimization of software reliability growth model with S-shaped testing-effort function using improved swarm intelligent optimization," *Appl. Soft Comput.*, vol. 40, pp. 283–291, Mar. 2016, doi: 10.1016/j.asoc.2015.11.041.
[44] C.-Y. Huang and M. R. Lyu, "Optimal release time for software systems considering cost, testing-effort, and test efficiency," *IEEE Trans. Rel.*, vol. 54, no. 4, pp. 583–591, Dec. 2005, doi: 10.1109/TR.2005.859230.
[45] C. Zhang, G. Cui, H. Liu, F. Meng, and S. Wu, "A unified and flexible framework of imperfect debugging dependent SRGMs with testing-effort," *J. Multimedia*, vol. 9, no. 2, pp. 310–317, Feb. 2014.
[46] 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, doi: 10.1504/IJRS.2013.057423.

[47] H. Pham, "Loglog fault-detection rate and testing coverage software reliability models subject to random environments," *Vietnam J. Comput. Sci.*, vol. 1, no. 1, pp. 39–45, Feb. 2014, doi: 10.1007/s40595-013-0003-4.
[48] J. H. Lo, "Toward a unified approach to software reliability modeling under imperfect debugging," *Appl. Mech. Mater.*, vols. 764–765, pp. 979–982, May 2015, doi: 10.4028/www.scientific.net/AMM.764-765.979.
[49] Q. Li and H. Pham, "A generalized software reliability growth model with consideration of the uncertainty of operating environments," *IEEE Access*, vol. 7, pp. 84253–84267, 2019, doi: 10.1109/ACCESS.2019.2924084.
[50] S. Yamada, *Software Reliability Modeling: Fundamentals and Applications*. Tokyo, Japan: Springer-Verlag, 2014, pp. 39–86, doi: 10.1007/978-4-431-54565-1.
[51] Z. Liu, S. Yang, M. Yang, and R. Kang, "Software belief reliability growth model based on uncertain differential equation," *IEEE Trans. Rel.*, vol. 71, no. 2, pp. 775–787, Jun. 2022, doi: 10.1109/TR.2022.3154770.
[52] B. Liu, "Fuzzy process, hybrid process and uncertain process," *J. Uncertain Syst.*, vol. 2, no. 1, pp. 3–16, 2008, doi: 10.1109/TR.2022.3154770.
[53] 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, doi: 10.1109/TR.2006.879611.
[54] Q. Li, C. Zhang, and H. Zhang, "A new software reliability model for open stochastic system based on NHPP," in *Proc. IEEE Int. Conf. Softw. Quality, Rel. Secur. Companion (QRS-C)*, Jul. 2017, pp. 624–625, doi: 10.1109/QRS-C.2017.120.
[55] C.-T. Lin and C.-Y. Huang, "Staffing level and cost analyses for software debugging activities through rate-based simulation approaches," *IEEE Trans. Rel.*, vol. 58, no. 4, pp. 711–724, Dec. 2009, doi: 10.1109/TR.2009.2019669.
[56] J.-S. Lin, C.-Y. Huang, and C.-C. Fang, "Analysis and assessment of software reliability modeling with preemptive priority queueing policy," *J. Syst. Softw.*, vol. 187, May 2022, Art. no. 111249, doi: 10.1016/j.jss.2022.111249.
[57] J.-S. Lin and C.-Y. Huang, "Queueing-based simulation for software reliability analysis," *IEEE Access*, vol. 10, pp. 107729–107747, 2022, doi: 10.1109/ACCESS.2022.3213271.
[58] T. Kim, K. Lee, and J. Baik, "An effective approach to estimating the parameters of software reliability growth models using a real-valued genetic algorithm," *J. Syst. Softw.*, vol. 102, pp. 134–144, Apr. 2015, doi: 10.1016/j.jss.2015.01.001.
[59] T. Yaghoobi, "Parameter optimization of software reliability models using improved differential evolution algorithm," *Math. Comput. Simul.*, vol. 177, pp. 46–62, Nov. 2020, doi: 10.1016/j.matcom.2020.04.003.
[60] L. Yang, Z. Li, D. Wang, H. Miao, and Z. Wang, "Software defects prediction based on hybrid particle swarm optimization and sparrow search algorithm," *IEEE Access*, vol. 9, pp. 60865–60879, 2021, doi: 10.1109/ACCESS.2021.3072993.
[61] R. Garg, S. Raheja, and R. K. Garg, "Decision support system for optimal selection of software reliability growth models using a hybrid approach," *IEEE Trans. Rel.*, vol. 71, no. 1, pp. 149–161, Mar. 2022, doi: 10.1109/TR.2021.3104232.
[62] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood, "Evaluation of competing software reliability predictions," *IEEE Trans. Softw. Eng.*, vol. SE-12, no. 9, pp. 950–967, Sep. 1986, doi: 10.1109/TSE.1986.6313050.
[63] Y.-F. Hou, C.-Y. Hung, and C.-C. Fang, "Using the methods of statistical data analysis to improve the trustworthiness of software reliability modeling," *IEEE Access*, vol. 10, pp. 25358–25375, 2022, doi: 10.1109/ACCESS.2022.3154103.
[64] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*. New York, NY, USA: McGraw-Hill, 1987, pp. 315–317.

HANG LUO received the B.S. degree in physical education from Sichuan Normal University, China, in 1995, and the M.S. degree in radio physics and the Ph.D. degree in detection technology and automation equipment from the University of Electronic Science and Technology of China (UESTC), in 2004 and 2010, respectively. He was a Postdoctoral Fellow, in 2014, and a Visiting Scholar, in 2018, with the School of Computer Science, McGill University, engaging in software reliability research. He is currently with the School of Mechanical Engineering, Sichuan University, as an Assistant Professor. His research interests include modeling of software reliability, reliability analysis (in the area of fault trees), and theory and application of adaptive and intelligent control.

LIJIA XU received the master's degree from the Beijing University of Technology, in 2004, and the Ph.D. degree in automation engineering from the University of Electronic Science and Technology of China, in 2009. Now, she is a Professor with Sichuan Agriculture University. Her main research interests include intelligent signal processing, intelligent algorithm, and fault diagnosis.

LIANG HE received the Ph.D. degree from Tohoku University, Japan, in 2013. He is currently a Professor with the School of Mechanical Engineering, Sichuan University. His current research interests include the micromachining of micro-electromechanical systems (MEMSs), integrated micro-device, and reliability.

LANDU JIANG received the B.Eng. degree in information security engineering from Shanghai Jiao Tong University, the M.Sc. degree in computer science from the University of Nebraska–Lincoln, the master's degree (minor) in construction management, and the Ph.D. degree from the School of Computer Science, McGill University. His research interests include computer vision, machine learning, smart sensing, wearable and mobile computing, cyber-physical systems, green energy solutions, online social networks, and software reliability.

TING LONG received the master's degree in electronic information engineering from Tianjin University, in 2007, and the Ph.D. degree in detection technology and automation equipment from the University of Electronic Science and Technology of China, in 2011. Now, she is an Associate Professor with the Sichuan Post and Telecommunications College. Her main research interests include application mode, information processing of the Internet of Things, and reliability.