Eksploatacja i Niezawodnosc – Maintenance and Reliability

Volume 25 (2023), Issue 2

journal homepage: http://www.ein.org.pl

Article citation info:
Zheng Z, Yang J, Hu Y, Wang X, Open-source Software Reliability Modeling with Stochastic Impulsive Differential Equations Eksploatacja i Niezawodnosc – Maintenance and Reliability 2023: 25(2) http://doi.org/10.17531/ein/166342

Open-source Software Reliability Modeling with Stochastic Impulsive Differential Equations

Indexed by: Web of Science Group

Zhoutao Zheng[a], Jianfeng Yang[a,b,*], Yao Hu[a], Xibing Wang[b]

[a] School of Mathematics and Statistics, Guizhou University, China
[b] School of Data Science, Guizhou Institute of Technology, China

Highlights

- Reliability modeling with stochastic impulsive differential equations (SIDE) is proposed.
- The model divides dynamic process of software fault into a continuous and a skipped part.
- The proposed model with SIDE is more in line with reality and has a better fitting effect.

Abstract

In reality, sudden updates of software, attacks of hackers, influence of the Internet market, etc. can cause a surge in the number of open-source software (OSS) faults (this moment is the time when impulse occurs), which results in impulsive phenomenon. For the existing software reliability models, dynamic process of software fault is considered to be continuous when assessing reliability, but continuity of the process can be disrupted with appearance of random impulses. Thus, to more accurately assess software reliability, we proposed an OSS reliability model with SIDE. In the model, dynamic process of software fault is divided into a continuous and a skipped part, described the continuous part of the process with SDE, and described destruction of the continuity caused by unpredictable random events with random impulses. Finally, the proposed model is verified with two datasets from real OSS project, and the results show that the proposed model is more in line with reality and has better fitting effect than the existing models.

Keywords

open-source software, reliability modeling, random impulse, stochastic impulsive differential equations

1. Introduction

In today's information age, with the increasing use of computers, the consequences of computer faults are becoming increasingly serious. Once a fault occurs, it might lead to major losses. Thus, the quality of the software needs to be improved urgently. Software reliability as an important evaluation attribute of software quality has been widely studied. Software reliability modeling analysis is an effective means to improve software reliability. It is well known that completeness, accuracy, and consistency of reliability assessment are important measurable criteria in reliability modeling. However, a novel and easy way has been chosen to describe the metric of reliability assessment in these books [16, 17], that is a Hausdorff metric was chosen to evaluate the test data which are fitted to the sigmoid models proposed. In the past, a number of classical traditional software reliability growth models [6, 15, 27, 29, 30] and imperfect debugging models [8, 19, 24, 25] have been proposed and widely applied. In these models, the fault detection rate is described by power function, S-type function or exponential

(*) Corresponding author.
E-mail addresses: Z. Zheng (ORCID: 0000-0002-9211-3621) ztzheng.gzu@qq.com, J. Yang (ORCID: 0000-0003-3486-9604) yjf@git.edu.cn, Y. Hu yhu1@gzu.edu.cn, X. Wang (ORCID: 0000-0003-1831-9872) binxiwang@git.edu.cn,

function. However, the testing efforts spent in the actual testing process, the skill level of testers and various testing tools will affect the fault and reliability of the software in an unpredictable way, resulting in a certain random impact on the detection rate of each fault, which may produce irregular fluctuations. Aim at the influence brought by these uncertain factors, some scholars solve the problem by adding noise term into $b(t)$ [3, 4, 14, 23, 26]; Other scholars build a data-driven reliability model [5, 13, 22] based on the data itself. This kind of model can interpret and analyze the fault data very well, but it can't establish a specific mathematical model analytic formula, which makes it difficult for managers to adjust and analyze the software fault later.
Furthermore, when the influence fluctuation is too large, it will have a great impact on the average behavior of the software fault detection process in the test stage. Therefore, how to build an open-source software (OSS) reliability model in line with reality is a worthy study. To solve the above problems, a software reliability model with stochastic impulsive differential equations is proposed. That is to say, combined with the actual analysis, when the random impact of major damage on product reliability in the process of software testing is encountered, and then the dynamic process of software fault can be divided into continuous part and skip part. Stochastic differential equations are used to describe the continuous part of the fault process, and stochastic impulse are used to describe the damage of the continuity caused by unpredictable random major events. The software reliability model with SIDE is considered to have better fitting effect than the traditional model and the software reliability models with stochastic differential equations [7].
The remainder of this paper has been organized as follows: The preparatory knowledge is presented in Section 2. The theoretical derivation of the proposed SRGM with stochastic impulsive differential equations is given in Section 3. The estimation of the impulsive times and the estimation of the model parameters is described in Section 4. The numerical validation of the proposed method performed by using two real fault datasets of Firefox and R is presented in Section 5. The study is summarized in Section 6, along with a discussion of the proposed model.

2. Preparatory Knowledge

2.1 Brownian Motion

I. Brownian motion

A stochastic process $\{B(t), t \geq 0\}$ called a Wiener process (or Brownian motion), if

(i) $B(0) = 0$,

(ii) $\{B(t), t \geq 0\}$ is a process with stationary independent increments, and

(iii) $\forall t > 0, B(t) \sim N(0, \sigma^2 t)$.

It should be noted that when $\sigma = 1, \{B(t), t \geq 0\}$ is called the standard Brownian motion. The condition in (i) is not necessary. If $B(0) = x$, then $\{B(t), t \geq 0\}$ is called the Brownian motion starting with x and is denoted as $B^x(t)$, and $B^x(t) \sim N(x, \sigma^2)$.
Such processes are often used for describing random noise.

II. Geometric Brownian Motion

If

$$X(t) = e^{sB(t)}, t \geq 0.$$

Then the stochastic process $\{X(t), t \geq 0\}$ is called the geometric Brownian motion.

i. Matrix function of the Brownian motion

The matrix function of $B(t) \sim N(0, t)$:

$$M_B(t) = E[e^{sB(t)}] = e^{\frac{s^2 t}{2}}.$$

ii. Mean function and variance function

The mean function and the variance function of the geometric Brownian motion can be calculated by the moment matrix function of the Brownian motion as follows:

$$E[X(t)] = E[e^{sB(t)}] = e^{\frac{s^2 t}{2}},$$
$$\mathrm{var}[X(t)] = E[X^2(t)] - (E[X(t)])^2 = E[e^{2sB(t)}] - e^{s^2 t} = e^{2s^2 t} - e^{s^2 t}. \quad (2.1)$$

2.2 Newtonian Algorithm

In Section 4 of this paper, we need to perform the necessary parameter estimates for the unknown parameters of the proposed model. To achieve the evaluation goal, it is necessary to solve the nonlinear equations. Therefore, in this section, we introduce the solution of the nonlinear equations in order to realize parameter estimation, such as solving the extreme value of the likelihood function using the Davidon-Fletcher-Powell (DFP) algorithm and the quasi-Newtonian method [9].
A nonlinear equation can be solved by constructing a quasi-

Newtonian iteration method. This method not only ensures fast convergence of Newton's method but also reduces the amount of calculation. The basic idea of this method involves the construction of an approximation of the Hesse matrix by using the difference in the gradient of the objective function and then generating a search direction based on Newton's equation. Finally, the iterative process is completed by a line search. Using the quasi-Newtonian method, the extreme value of the likelihood function, $f(X)$, is solved and the value of the variable X is found when it is taken to the extremum. The flowchart of the algorithm is as follows:

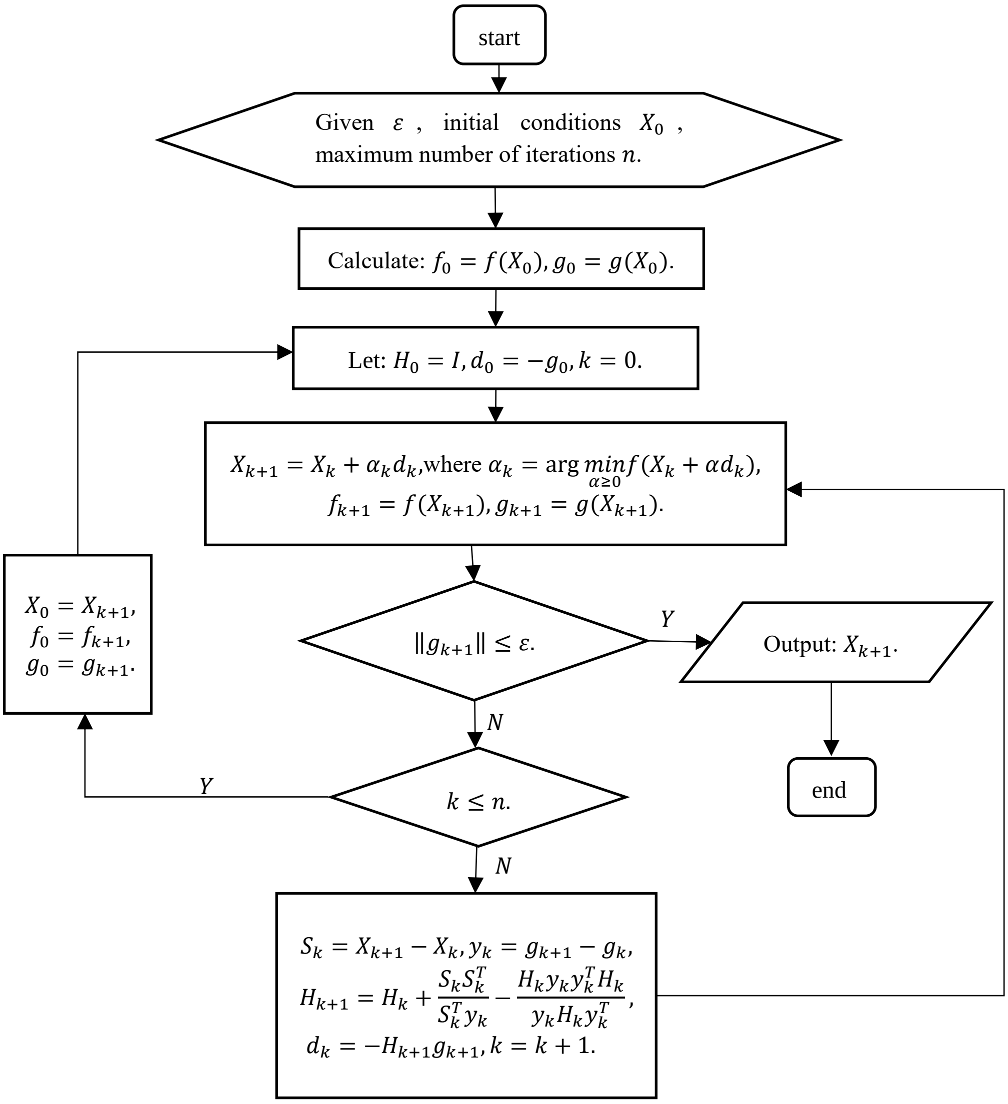

Figure 1. Algorithm flow chart.

3. Open-Source Software Reliability Modeling with Stochastic Impulsive Differential Equations

3.1 Modeling Preparation and Assumptions

In some stages of system development, the mutation phenomenon corresponding to a rapid change in the system due to the interference of external factors is usually called the impulsive phenomenon, and the location of the change is called the impulsive time (or the abrupt change-point). This process is generally very short. For instance, the trend of the stock market is affected by the phenomenon of the occurrence of a random impulse when big news arrives. In this study, the number of software faults discussed will be affected by the random impulsive phenomenon, resulting in unrealistic software reliability of the previous deterministic modeling.

$N(t)$ is taken to be the cumulative number of detection faults of the software at the test time $t (t \geq 0)$ and the following assumptions are made:

(1) The dynamic change in the detection of the number of faults can be described by the geometric Brownian motion.
(2) In the test environment, some factors can cause the number of faults to surge.

Under common assumptions for software reliability growth modeling, the following linear differential equation can be obtained,

$$\frac{d(a-N(t))}{dt} = -b(t)(a-N(t)) \quad (3.1)$$

Let

$$M(t) = a - N(t).$$

Equation (3.1) is equivalent to the following expression,

$$\frac{dM(t)}{dt} = -b(t)M(t). \quad (3.2)$$

Where a is the total number of initials in the software, $b(t)$ represents the fault detection rate function, which implies the probability that each latent fault in the product is detected at time t, $M(t)$ represents the number of remaining faults of the system at time t.

Based on the above assumption (1) and common assumptions for software reliability growth modeling, considering the irregular motion state of software faults, the number of software faults during the software testing phase is similar to the random fluctuations of stock price changes, and the occurrence of faults varies with time. Thus, the following SDE is obtained by extending Equation (3.2).

$$\frac{dM(t)}{dt} = -[b(t) - \sigma\gamma(t)]M(t). \quad (3.3)$$

Where $\gamma(t)$ is the standardized Gaussian white noise; σ is a positive constant, and

$$E[\gamma(t)] = 0, V[\gamma(t)] = \sigma^2.$$

Mathematically, the Wiener process $B(t)$ can describe the integral form of the Gaussian white noise [12], i.e.

$$B(t) = \int \gamma(t)dt,$$
$$dB(t) = \gamma(t)dt.$$

And thus, Equation (3.3) is extended to an $I\hat{t}o$-type SDE as follows [23],

$$dM(t) = -[b(t) - \frac{1}{2}\sigma^2]M(t)dt - \sigma M(t)dB(t). \quad (3.4)$$

According to the $I\hat{t}o$ lemma [2], the solution of Equation (3.4) can be obtained as follows:

$$M(t) = a \cdot \exp[-\int_0^t b(s)ds - \sigma B(t)]. \quad (3.5)$$

According to equation (3.2),

$$N(t) = a \cdot \{1 - \exp[-\int_0^t b(s)ds - \sigma B(t)]\}. \quad (3.6)$$

For the detailed derivation process of the conclusions (3.5), (3.6), please refer to Derivation-A.

From Equation (3.6) and the definition of the geometric Brownian motion, it can be seen that $N(t)$ represents a geometric Brownian motion. From the properties of the geometric Brownian motion, the progressive properties of $N(t)$ can be obtained,

Because $-\int_0^t b(s)ds < 0$, when $t \to \infty$, we get

$$\exp[-\int_0^t b(s)ds - \sigma B(t)] \overset{t\to\infty}{\to} 0.$$

Thus,

$$\lim_{t\to\infty} N(t) = a.$$

Because $\{B(t), t \geq 0\}$ is a Wiener process, according to its properties, we get

$$B(t) \sim N(0, \sigma^2 t).$$

Based on the properties of the geometric Brownian motion and Brownian motion, $M(t), N(t)$ follows a logarithmic normal distribution, and their distribution function is obtained as follows [26],

$$P[M(t) \leq m | M(0) = a] = \Phi(\frac{\log(\frac{m}{a}) + \int_0^t b(s)ds}{\sigma\sqrt{t}}). \quad (3.7)$$

$$P[N(t) \leq n | N(0) = 0, M(0) = a] =$$
$$\Phi(\frac{\log(\frac{a}{a-n}) + \int_0^t b(s)ds}{\sigma\sqrt{t}})(n < a). \quad (3.8)$$

The function $\Phi(\cdot)$ in Equations (3.7), (3.8) is the standardized normal distribution function which is defined as follows:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{s^2}{2}\right) ds. \quad (3.9)$$

At a given time t, the stochastic process $M(t), N(t)$ is a random variable, the mathematical expectation of which can be calculated. From Equations (2.1), (3.5),(3.6), the mathematical expectation and variance of the process $M(t), N(t)$ at time t can be calculated as follows:

$$E[M(t)] = a \cdot \exp[-\int_0^t b(s)ds + \frac{\sigma^2}{2}t],$$
$$m(t) = E[N(t)] = a \cdot \{1 - \exp[-\int_0^t b(s)ds + \frac{\sigma^2}{2}t]\},$$
$$\mathrm{var}[M(t)] = \mathrm{var}[N(t)] = a^2 \cdot \exp\left[-2\int_0^t b(s)ds + \sigma^2 t\right] \cdot [\exp(\sigma^2 t) - 1]. \quad (3.10)$$

3.2 Reliability Modeling with Stochastic Impulsive Differential Equations

In realistic situations, we know that the reliability of software testing depends heavily on the community members and users. In addition, the detected number of faults can fluctuate greatly when some significant news (such as the social development trend, the expert's advice, and software upgrade) comes along, which is identified as an impulsive time problem. The impulsive problem is considered based on the existing SRGM using SDE [26, 28]. Based on the above assumptions, in order to deal with the effects caused by such fluctuations and to be able to characterize the software operational profile precisely, we proposed the following SRGM with stochastic impulsive differential equations,

$$\begin{cases} dM(t) = -[b_i(t) - \frac{1}{2}\sigma_i^2]M(t)dt - \sigma_i M(t)dB(t), t \in [\tau_{i-1}, \tau_i) \\ M(\tau_i) = \eta_i M(\tau_i^-), i = 1,2,\dots \\ M(t) = a - N(t) \end{cases} \tag{3.11}$$

Where $\{B(t), t \geq 0\}$ is the standard Brownian motion, $\tau_1 \leq \tau_2 \leq \tau_3 \leq \cdots$ is a sequence of impulsive times, η_i is the impulsive function, $(\eta_i - 1)M(\tau_i^-)$ represents the jump amplitude of the number of faults, when the ith impulse occurs, $M(\tau_i)$ is the actual value at the time of the impulse, $M(\tau_i^-)$ is the value of $M(\tau_i)$ assuming that the impulse does not occur. The value of $M(\tau_i^-)$ can be obtained by simulating the geometric Brownian process. $\omega_i = \tau_i - \tau_{i-1}$ is a random series of variables for the wait interval between impulse occurrences, which obeys the two-parameter exponential distribution. For more theoretical knowledge of stochastic impulsive differential equations, the reader can refer to the literatures [11, 20, 21].

For n impulsive times, in solving Equation (3.11), it holds that

$$N(t) = \begin{cases} N_1(t), 0 \leq t < \tau_1 \\ N_2(t), \tau_1 \leq t < \tau_2 \\ \dots \\ N_n(t), \tau_{n-1} \leq t < \tau_n \\ N_{n+1}(t), t \geq \tau_n \end{cases} \tag{3.12}$$

The fault detection rate is given by

$$b(t) = \begin{cases} b_1(t) = b_1, 0 \leq t < \tau_1 \\ b_2(t) = b_2, \tau_1 \leq t < \tau_2 \\ \dots \\ b_n(t) = b_n, \tau_{n-1} \leq t < \tau_n \\ b_{n+1}(t) = b_{n+1}, t \geq \tau_n \end{cases} \tag{3.13}$$

The mean value function can be obtained using its computational properties,

$$m(t) = \begin{cases} m_1(t), 0 \leq t < \tau_1 \\ m_2(t), \tau_1 \leq t < \tau_2 \\ \dots \\ m_n(t), \tau_{n-1} \leq t < \tau_n \\ m_{n+1}(t), t \geq \tau_n \end{cases} \tag{3.14}$$

Where

$$m_i(t) = E[N_i(t)].$$

Using equation (3.10) we can get the following form of $m(t)$,

$$m(t) = \begin{cases} m_1(t) = a \cdot \left[1 - \exp\left(-(b_1 t) + \frac{1}{2}\sigma_1^2 t\right)\right], 0 \leq t < \tau_1 \\ m_2(t) = (a - m_1(\tau_1)) \cdot [1 - \exp(-\left(b_2(t - \tau_1)\right) \\ \qquad + \frac{1}{2}\sigma_2^2(t - \tau_1))] + m_1(\tau_1), \tau_1 \leq t < \tau_2 \\ \dots \\ m_{n+1}(t) = (a - m_n(\tau_n)) \cdot [1 - \exp(-(b_{n+1}(t - \tau_n)) \\ \qquad + \frac{1}{2}\sigma_{n+1}^2(t - \tau_n))] + m_n(\tau_n), t \geq \tau_n \end{cases} \tag{3.15}$$

4. Parameter Estimation with Random Impulse

4.1 Estimation Method of Impulsive Time

According to assumption (1), the number of faults, $W(t)$, between adjacent intervals when impulses occur, follows the geometric Brownian motion. The impulsive time is estimated by the properties of the geometric Brownian motion and maximum likelihood estimation. Assume that there are n groups of data containing the detected faults, $\{W_i, i = 1,2,\dots,n\}$, where W_i is the number of faults in time $[i-1, i]$, and $W_i = N(i) - N(i-1)$. Let the logarithmic growth rate be,

$$U_i = \log(W_{i+1}) - \log(W_i), i = 1,2,\dots,n-1.$$

It follows from the properties of geometric Brownian motion that

$$U_i \sim N(E[(U_i)], V[(U_i)]).$$

According to the normality of U_i, the impulsive time point of the number of faults can be estimated using the maximum likelihood method by performing the following steps:

(i) The normal distribution test function in MATLAB is used to test whether $U = \{U_1, U_2, \dots, U_n\}$ follows the normal distribution. If the set of data does not follow the normal distribution, there is at least one impulsive time in the sequence.

(ii) If an impulsive time exists, it is located using the maximum likelihood method.

$$L_k = l_1(k) \cdot l_2(k), k = 2,3,\dots,n-2. \tag{4.1}$$

Where

$$l_1(k) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi S_1^2(k)}} \exp(-\frac{(U_i - \overline{U_1(k)})^2}{2S_1^2(k)}),$$

$$\overline{U_1(k)} = \frac{1}{k}\sum_{i=1}^{k} U_i, S_1^2(k) = \frac{1}{k-1}\sum_{i=1}^{k}(U_i - \overline{U_1(k)})^2, k \geq 2,$$

$$l_2(k) = \prod_{i=k+1}^{n} \frac{1}{\sqrt{2\pi S_2^2(k)}} \exp(-\frac{(U_i - \overline{U_2(k)})^2}{2S_2^2(k)}),$$

$$\overline{U_2(k)} = \frac{1}{n-k}\sum_{i=k+1}^{n} U_i, S_2^2(k) = \frac{1}{n-k-1}\sum_{i=k+1}^{n}(U_i - \overline{U_2(k)})^2, k \leq n-2.$$

Log-transforming the equation (4.1), we get

$$\log L_k = -\frac{k}{2}\log(2\pi S_1^2(k)) + \frac{1}{2S_1^2(k)}\sum_{i=1}^{k}(U_i - \overline{U_1(k)})^2 + \frac{n-k}{2}\log(2\pi S_2^2(k)) + \frac{1}{2S_2^2(k)}\sum_{i=k+1}^{n}(U_i - \overline{U_2(k)})^2. \quad (4.2)$$

Obviously, the log-likelihood function given by Equation (4.2) can be regarded as a function of the variable k. The value of k for which the log-likelihood function is maximized is calculated. Here, k is denoted as $\tau = k(1 \leq k \leq n)$, i.e., an impulsive time of U. Next, the entire dataset is divided into two sub-sequences: $U_1, U_2, \ldots, U_k$ and $U_{k+1}, U_{k+2}, \ldots, U_n$.

(iii) Steps (i) and (ii) are repeated for $U_1, U_2, \ldots, U_k$ and $U_{k+1}, U_{k+2}, \ldots, U_n$, respectively, and for each of their sub-sequences, until none of the sub-sequences contain impulsive time, i.e., each sub-sequence obeys the normal distribution.

4.2 Maximum Likelihood Estimation

Considering the data between two adjacent impulsive times as a group, the entire sequence is divided into $n+1$ group data. Writing their likelihood functions, and the values of $a, b = (b_1, b_2, \ldots, b_{n+1}), \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_{n+1})$ are unknown parameters in Equation (3.15) are estimated using the maximum likelihood estimation method.

Taking the $i = 1$ group of data as an example, estimation of the unknown parameters corresponding to the other groups can be obtained using the same method. $(t_j, m_j)(j = 1,2,\ldots,K_1; 0 < t_1 - \tau_{i-1} < t_2 - \tau_{i-1} < \cdots < t_{K_1} - \tau_{i-1})$, where $\tau_0 = 0$ is the number of n detected faults, m_j represents the number of detected faults on $(\tau_{i-1}, \tau_i]$, and $m_0 = 0$. K_i represents the data observed in the group i. From Equation (3.8) and considering that $N(t)$ has Markov property, let the likelihood function of the process $N(t)$ be denoted as

$$L(a, b_1, \sigma_1) = P[N(t_1) \leq m_1, N(t_{K_i}) \leq m_{K_i} | N(0) = 0, M(0) = a - m(\tau_{i-1})]$$
$$= \prod_{j=1}^{K_1} P[N(t_j) \leq m_j | N(t_{j-1}) \leq m_{j-1}]$$
$$= \Phi\left(\frac{\log(\frac{a-m_{j-1}}{a-m_j}) - b_i \cdot (t_j - t_{j-1})}{\sigma_1\sqrt{t_j - t_{j-1}}}\right)$$
$$= \prod_{j=1}^{K_1} \frac{1}{((a-m_j)\sigma_1\sqrt{2\pi(t_j - t_{j-1})})} \cdot \exp\{-\frac{[\log(\frac{a-m_{j-1}}{a-m_j}) - b_1 \cdot (t_j - t_{j-1})]^2}{2\sigma_1^2(t_j - t_{j-1})}\}. \quad (4.3)$$

Logarithmic transformation of the likelihood Equation (4.3) to obtain the following log-likelihood function,

$$l = \log L(a, b_1, \sigma_1)$$
$$= -K_1 \log\sigma_1 - \sum_{j=1}^{K_1}\log(a - m_j) - \frac{1}{2}\sum_{j=1}^{K_1}\log(t_j - t_{j-1}) - \frac{1}{2\sigma_1^2}\sum_{j=1}^{K_1}\frac{\left[\log\frac{a-m_{j-1}}{a-m_j} - b_1 \cdot (t_j - t_{j-1})\right]^2}{(t_j - t_{j-1})} - \frac{K_1}{2}\log 2\pi. \quad (4.4)$$

The maximum likelihood estimation can be obtained as the solutions of the following simultaneous likelihood equations by using the Quasi Newtonian algorithm,

$$\frac{\partial l}{\partial \sigma_1} = -\frac{K_1}{\sigma_1} + \frac{1}{\sigma_1^3} \cdot \sum_{j=1}^{K_1}\frac{\left[\log(\frac{a-m_{j-1}}{a-m_j}) - b_1 \cdot (t_j - t_{j-1})\right]^2}{(t_j - t_{j-1})} = 0,$$
$$\frac{\partial l}{\partial a} = -\sum_{j=1}^{K_1}\frac{1}{a-m_j} + \frac{1}{\sigma_1^2} \cdot \sum_{j=1}^{K_1}\frac{(m_j - m_{j-1}) \cdot \left[\log(\frac{a-m_{j-1}}{a-m_j}) - b_1 \cdot (t_j - t_{j-1})\right]}{(t_j - t_{j-1}) \cdot (a - m_j) \cdot (a - m_{j-1})} = 0,$$
$$\frac{\partial l}{\partial b_1} = \frac{1}{\sigma_1^2} \cdot \sum_{j=1}^{K_1}\left[\log\left(\frac{a-m_{j-1}}{a-m_j}\right) - b_1 \cdot (t_j - t_{j-1})\right] = \frac{1}{\sigma_1^2} \cdot [\log(\frac{a}{a-m_{K_1}}) - b_1 t_{K_1}] = 0. \quad (4.5)$$

5. Experimental Data and Performance Analysis

5.1 Performance Evaluation Criteria

To do a fair comparison with the performance of various models, we used the following comparison criteria:

(i) Mean Square Error (MSE)

The mean square error (MSE) is generally defined as

$$MSE = \frac{1}{n} \cdot \sum_{k=1}^{n}[m(t_k) - m_k]^2. \quad (5.1)$$

Where n is the size of the selected dataset, m_k is the true number of faults at times t_k, and $m(t_k)$ is the estimated number of faults at times t_k.

A smaller value of MSE represents a minimum fitting error and thus, a better the model.

(ii) Akaike Information Criterion (AIC)

A smaller value of the Akaike information criterion (AIC) indicates a minimum fitting error, and thus, a better model. The AIC is generally defined as

$$AIC = 2K - 2\log(L). \quad (5.2)$$

Where K is the number of parameters, and L is the maximum likelihood value.

If the error of the model is normally distributed, then the AIC is defined as

$$AIC = 2K + n\log(\frac{RSS}{n}). \quad (5.3)$$

Where n is the size of the selected dataset, and RSS is the residual sum of squares.

(iii) Residual Error (RE)

The residual error (RE) is a common criterion used for judging the validity of the model prediction. Its specific definition is as follows,

$$RE = \frac{m(t_k)-m_k}{m_k}. \quad (5.4)$$

where m_k is the true number of faults at times t_k, and $m(t_k)$ is the estimated number of faults at times t_k. The closer the value of RE is to 0, the more accurate is the model prediction is.

5.2 Analysis of Actual Data

5.2.1 Description of The Actual Dataset

We focused on the Firefox browser and R software which are the software systems that have been developed under the open-source project. According to the method proposed in section 3, two fault data corresponding to the Firefox browser and R, taking from the website https://www.bugzilla.org/, were used for the numerical verification. Dataset 1 corresponding to the Firefox browser, included 333 weeks of fault data from January 1, 2016, to May 19, 2022, as listed in Table 1.

Table 1. Actual fault dataset of Firefox.

Time	number	time	number	time	number	time	number	time	number
0	0	6	4	12	2	18	7	327	13
1	7	7	2	13	4	19	5	328	13
2	5	8	12	14	9	20	8	329	15
3	8	9	4	15	3	21	5	331	20
4	11	10	7	16	9	22	5	332	16
5	4	11	3	17	14	…	…	333	14

Table 2. Actual fault dataset of R.

Time	number	time	number	time	number	time	number	time	number
0	0	6	6	12	1	18	3	84	4
1	5	7	6	13	2	19	6	85	4
2	7	8	4	14	7	20	11	86	5
3	5	9	6	15	8	21	13	87	11
4	2	10	6	16	8	22	11	88	10
5	5	11	4	17	9	…	…	89	1

From the website https://bugs.r-project.org/,__dataset 2 corresponding to R, included 88 two-weeks of fault data from January 1, 2019, to May 23, 2022, as listed in Table 2. .

In this section, the model proposed in this work has been compared with the classical GO model and the reliability model with SDE. The unknown parameters in the model were evaluated using the maximum likelihood method. Table 3 summarizes the form of the fault detection rate and the mean value function of the selected model.

Table 3. The model summary.

Model	Description	fault detection rate	MVF
#1	GO model	$b(t) = b$	$m(t) = a \cdot (1 - e^{-bt})$
#2	SDE-based model	$b(t) = b$	$m(t) = a \cdot [1 - e^{(-bt+\sigma^2/2)}]$
#3	SIDE-based model	$b(t) = \begin{cases} b_1, 0 \le t < \tau_1 \\ b_2, \tau_1 \le t < \tau_2 \\ \cdots \\ b_{n+1}, t \ge \tau_n \end{cases}$	(3.15)

5.2.2 The Estimation of Impulsive Time

I. Dataset 1

Based on the method described in Section 4.1, the impulsive times of the data from the Firefox fault dataset were identified in Table 1, and the results obtained have been shown in Table 4.

Table 4. Segmentation of the normality test of dataset 1.

Data	Time	H	P-value
Ds-1	0-333	1	0.4149
DS-2	0-103	0	0.3149
Ds-3	105-260	0	0.2573
Ds-4	262-333	0	0.5000

In the process of segmentation, it was found that the length of some of the sub-sequences without impulsive times was very small, i.e., the interval between the two impulsive times was very small, and experience and practice show that there should be a time difference between the two impulsive times. Thus, the impulsive times that were too small at such impulsive moments were excluded from the evaluation. By performing the impulsive time extraction step described in Section 4.1, the sequence was divided into three segments, i.e., two impulsive times were found at $\tau_1 = 104, \tau_2 = 261$. From the assumptions of the above modeling process, we know that the fault data of two adjacent impulsive times obey a geometric Brownian motion, i.e., its value after taking the logarithm fluctuates about the 0-mean point. Meanwhile, from Figure 2, we can also roughly see the impulsive time position of the sequence.

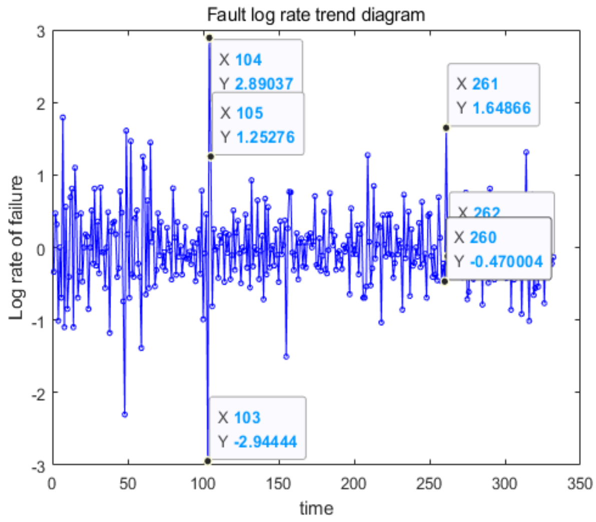

Figure 2. Log-rate change trend diagram showing the cumulative number of faults of Firefox.

Figure 2 is mainly drawn according to the changes of value of U_i in Section 4.1, representing the estimation of impulsive time for the fault data of Firefox mentioned above. And the results derived from them can be used as an initial value to solving for the exact impulsive time.

II. Dataset 2

The impulsive times of the data from the fault dataset of R were identified in Table 2. The sequence was divided into two segments, and the results obtained have been shown in Table 5, i.e., one impulsive time was found at $\tau_1 = 45$(or 46).

Table 5 Segmentation of the normality test of dataset 2

Data	Time/d	H	P-value
Ds-1	0-189	1	0.6456
Ds-2	0-44	0	0.5459
Ds-3	47-89	0	0.7361

From the assumptions of the above modeling process, we know that the fault data of two adjacent impulsive times obeys the geometric Brownian motion, i.e., its value after taking the logarithm fluctuates about the 0-mean point. At the same time, from Figure 3, we can also roughly see the impulsive time position of the sequence.

Figure 3 is mainly drawn according to the changes of value of U_i in Section 4.1, representing the estimation of impulsive time for the fault data of R mentioned above. And the results derived from them can be used as an initial value to solving for the exact impulsive time.

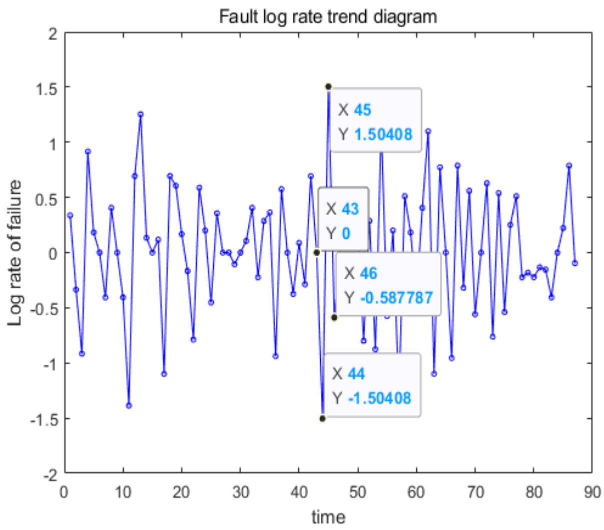

Figure 3. Log-rate change trend diagram showing the cumulative number of faults of R.

5.2.3 Comparison of The Performance Criteria

I. Dataset 1

The parameter estimates of the selected model obtained from the Firefox fault data in Table 1 are shown in the Table 6 . In addition, Table 7 shows the comparison of the values of MSE and AIC for these different models. From Table 7, we can see that the proposed reliability model with SIDEs is obviously better than the existing models that have been compared in this study.

Table 6 Parameter estimation of each model

Model	Description	$a(1\times10^4)$	$b(1\times10^{-1})$	Other
#1	NHPP-GO model	$a = 2.4697$	b=1.2500	
#2	SDE-based model	$a = 2.4725$	$b = 1.2451$	$\sigma = 0.0042$
#3	SIDE-based model	$a = 2.4668$	$b_1 = 0.5850$ $b_2 = 2.0001$ $b_3 = 0.6550$	$\sigma_1 = 0.0043$ $\sigma_2 = 0.0022$ $\sigma_3 = 0.0021$ $\tau_1 = 104$ $\tau_2 = 261$

Table 7 Model comparison results

Model	Description	MSE	$AIC(1\times10^3)$
#1	NHPP-GO model	127.5421	4.3190
#2	SDE-based model	124.9542	4.3142
#3	SIDE-based model	23.6821	3.7681

Figure 4 shows the relationship between the actual fault data and the predicted fault data of Firefox based on the traditional GO model, the SDE model, and the model proposed in this study. At the same time, a comparison of the fitting (prediction) results of the different models is shown, combined with the value of MSE and AIC in Table 7, it can be seen that there is little difference in the fitting effect between the traditional GO model and the SDE model, but their fitting effect is worse than the proposed SIDE model. And it is easily observed that the fault data predicted by the SIDE model is closer to the actual data from Figure 4.

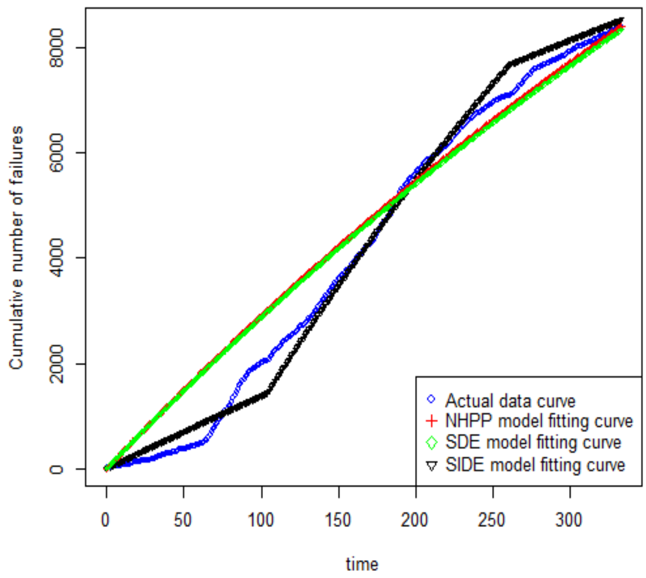

Figure 4. Comparison of the fitting results obtained from the different models considered.

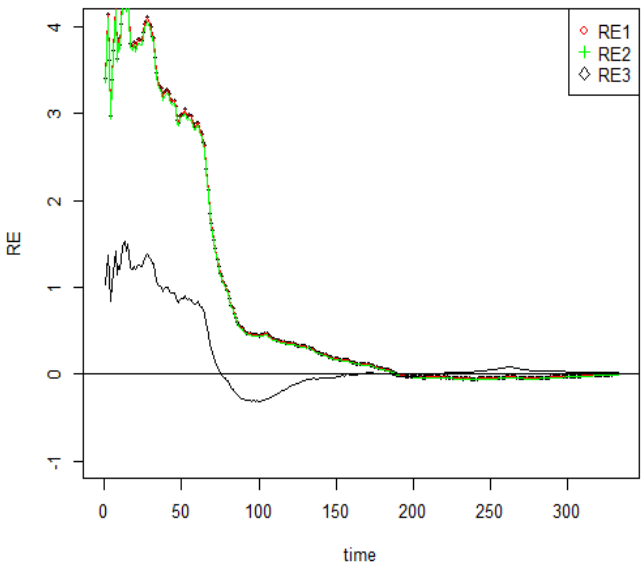

Figure 5. Comparison of the residuals of the three models.

Figure 5 presents a comparison chart of their residual errors, from which it can be seen that the values of RE are approaching 0 after time $t = 150$, indicating that all the mentioned models have good fitting results. It can be clearly seen that closest to 0 is of model #3, followed by that of model #2, and finally model #1. The closer the value of RE is to 0, the better the fit of the model is, which is consistent with the results in Figure 4 and Table 7. Thus, it can be seen from these figures and tables that the proposed reliability model with stochastic impulsive differential equations gives the best fit to the Firefox fault data in Table 1.

II. Dataset 2

As can be seen from Table 8 , the parameter estimates of the selected model are obtained using the R fault data given in Table 2. Further, Table 9 shows the comparison of the values of MSE and AIC for the different models, from which it can be seen that the proposed reliability model with the stochastic impulsive differential equations is obviously better than the existing models being compared.

Table 8 Parameter estimation of each model

Model	Description	$a(1\times10^3)$	$b(1\times10^{-3})$	other
#1	NHPP-GO model	$a = 1.8121$	$b = 6.8501$	
#2	SDE-based model	$a = 1.7001$	$b = 7.0600$	$\sigma = 0.0052$
#3	SIDE-based model	$a = 1.7890$	$b_1 = 4.9601$ $b_2 = 8.7800$	$\sigma_1 = 0.0030$ $\sigma_2 = 0.0044$ $\tau_1 = 45$(or 46)

Table 9. Model comparison results.

Model	Description	MSE	$AIC(1\times10^2)$
#1	NHPP-GO model	62.4941	7.7152
#2	SDE-based model	49.6070	7.5296
#3	SIDE-based model	16.7982	6.6058

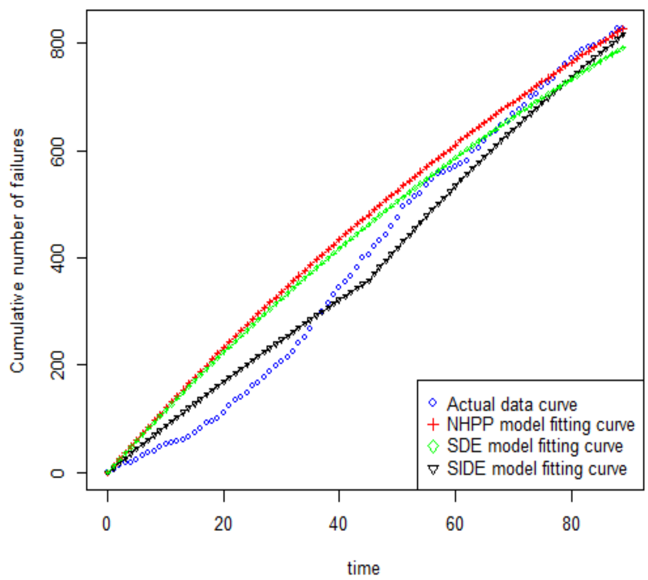

Figure 6. Comparison of the fitting results of the different models.

Figure 6 show the relationship between the actual fault data of R and the predicted fault data based on the traditional GO model, the SDE model, and the model proposed in this work. At the same time, a comparison of the fitting (prediction) results of the different models is shown, combined with the value of MSE and AIC in Table 9, it can be seen that there is little difference in the fitting effect between the traditional GO model and the SDE model, but their fitting effect is worse than the proposed SIDE model. And it can be seen that the fault data predicted by the SIDE model is closer to the actual data.

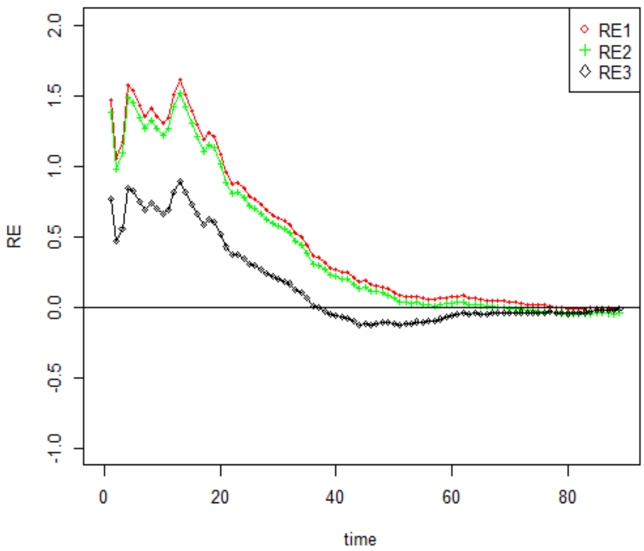

Figure 7. Comparison of the residuals of the different models.

Figure 7 presents a comparison chart of their residual errors, from which it can be seen that the values of RE are approaching 0 after time $t = 60$, indicating that all the mentioned models have good fitting results. From the figure, it can be clearly seen that the value of RE closest to 0 is that of model #3, followed by that of model #2, and finally model #1. The closer the value of RE is to 0, the better the fit of the model is, which is consistent with the results in Figure 6 and Table 9. As can be observed from these figures and tables, the proposed reliability model with the stochastic impulsive differential equations gives the best fit to the R fault data in Table 2. .

5.3 Sensitivity Analysis and Goodness of Fit

The purpose of sensitivity analysis is to investigate which parameter have important influence on the model. From Figure 8, we can see these parameters a, b have important influence on the proposed model, while $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ has little effect. The reasons are as follows:

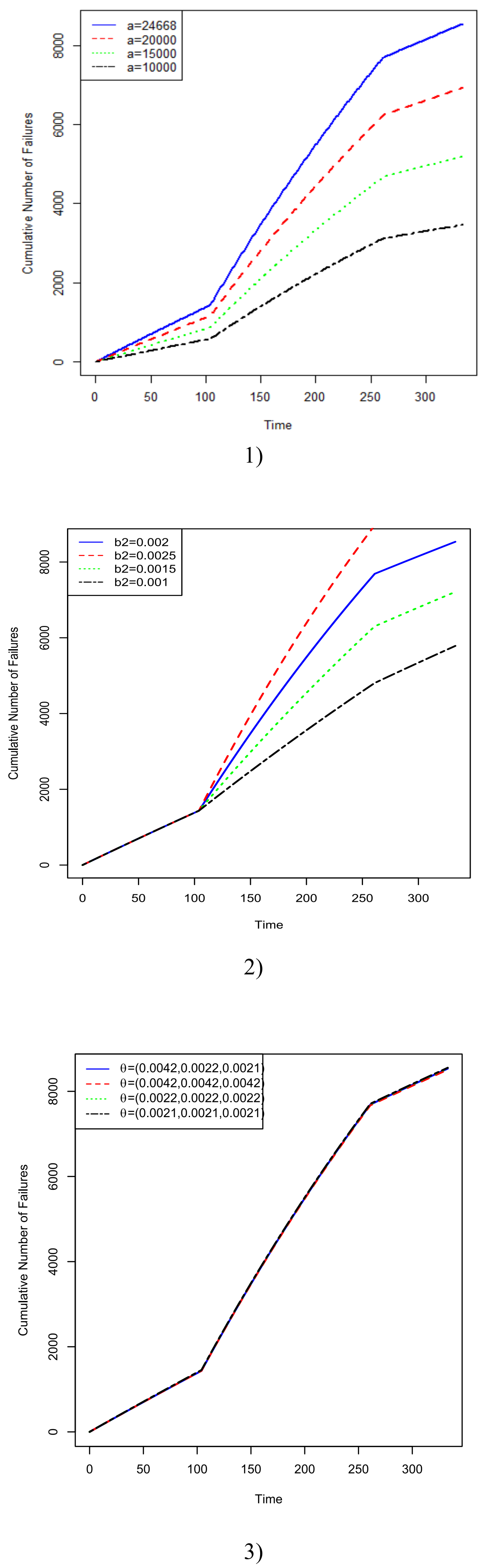

Figure 8. Sensitivity analysis of the proposed model parameters using dataset 1.

1) The total number of original faults (a) has an important impact in the process of the open-source software (OSS) development. Because the number of original faults directly affects and determines the quality and reliability of the OSS. It can be seen from (1) of Figure 8 that the total number of original faults has a great impact on the final fault fitting result. Therefore, this is a factor that must be considered when establishing software reliability model.
2) The fault detection rate (b) is also an important factor in the process of the OSS development and testing. It determines the probability of faults being detected in the OSS. Its change directly affects the number of faults detected in the OSS, but also indirectly affects the number of remaining faults in the OSS. It can be seen from (2) of Figure 8 that the change of the parameter b has a great influence on the final fault fitting result. Therefore, the parameter b must be considered in software reliability modeling.
3) It can be seen from (3) of Figure 8 that the fluctuation parameter σ has no significant influence on the final fault fitting result, because the whole fault detection process is separated and processed at impulsive times, which further makes the fluctuation of the fault detection is small in each section.

This paper focuses on the influence of random shocks (random impulses) on the modeling analysis of software reliability assessment, so the stochastic differential equations is used to solve the stochasticity in the modeling analysis. And because in the existing reliability models, the entire dynamic process of software fault is generally considered as continuous when evaluating software reliability, but in reality, due to the existence of some random shocks, the continuity of the dynamic process of software fault can be damaged. Therefore, according to the actual situation, the OSS reliability modeling with SIDE proposed in this paper is more in line with the reality and has better fitting effect than the existing software reliability models.

For the problem of "Overfitting" and "Underfitting" of the proposed model we can calculate its Hausdorff distance to characterize the goodness of fit of model by referring to literatures [10, 18]. For different models, the smaller the Hausdorff distance is, the better the model fitting effect will be, and an upper and lower bound on the Hausdorff distance can be calculated to better identify whether the model is " Overfitting " or " Underfitting".

6. Summary

In this study, we have presented the use of segmented geometric Brownian motion to describe the cumulative number of software faults, the use of the maximum likelihood estimation method to locate the impulsive times based on the properties of geometric Brownian motion, and to evaluate the parameters in the model. Because some major events will have an impact on the use of the software, resulting in a surge in the number of software faults and generating the impulsive phenomenon, the new model proposed in this study is closer to reality and provides an effective description of the software fault process. Finally, the proposed model, the NHPP-based model, and the reliability model with the SDE have been used to obtain fault predictions, and these have been compared with the actual fault data of Firefox and R obtained from https://www.bugzilla.org/. The comparison results show that the proposed model is more effective and practical than the existing models. From the figures and tables, it was observed that by dividing the data of dataset 1 into 3 segments and by querying the development history of Firefox, its two impulsive times corresponded to its major development history, i.e., the introduction of Firefox quantum on November 14, 2017, which greatly improved its performance, and the end of support for Adobe Flash in January 2021. The same is true for dataset 2.

The model proposed in this study can be seen as the influence of each major information on the software performance at a certain time (impact force), thereby improving the maintenance of the software in the impulsive time and more effectively describing the software fault process. It can be applied to the network maintenance of shopping apps at certain time (impulsive time) and the maintenance of other registration websites, etc.

Prospective development: In the future, we can consider the SRGM with SIDE incorporating user behavior + multi-version multi-mutation + masked data [1] and studied the intrinsic characteristic d- “supersaturation” [18] of new model, breaking through the global software fault elimination, which is conducive to software developers for faster debugging and upgrading the software to achieve improved software reliability.

Derivation-A.

$$\frac{dM(t)}{dt} = -b(t)M(t), M(t) = a - N(t). \tag{A.1}$$

Add random noise term, and the following SDE is obtained by extending Equation (A.1).

$$\frac{dM(t)}{dt} = -[b(t) + \sigma\gamma(t)]M(t). \tag{A.2}$$

Where $\gamma(t)$ is the standardized Gaussian white noise, and

$$\begin{aligned} &E[\gamma(t)] = 0, V[\gamma(t)] = \sigma^2, \\ &B(t) = \int \gamma(t)dt, \\ &dB(t) = \gamma(t)dt. \end{aligned} \tag{A.3}$$

Here $B(t)$ is a Gaussian process (Wiener process) with zero mean.

Putting equation (A.3) into equation (A.2), we can get the following derivation:

$$\begin{aligned} dM(t) &= -[b(t) + \sigma\gamma(t)]M(t)dt \\ &= -b(t)M(t)dt - \sigma M(t)\gamma(t)dt \\ &= -b(t)M(t)dt - \sigma M(t)dB(t). \end{aligned} \tag{A.4}$$

Apply the $I\hat{t}o$ lemma to solve the above SDE formula (A.4).

Let $f\big(t, B(t)\big) = \log\big(M(t)\big)$, From the $I\hat{t}o$ lemma, we can get

$$\begin{aligned} &d\log\big(M(t)\big) = df \\ &= \frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial B(t)}dB(t) + \frac{1}{2}\frac{\partial^2 f}{\partial B(t)^2}\big(dB(t)\big)^2 \\ &= \frac{1}{M(t)}dM(t) = -b(t)dt - \sigma dB(t). \end{aligned} \tag{A.5}$$

Integrate both sides of equation (A.5) at the same time, and finally take the logarithm on both sides of the equation.

$$\begin{aligned} &\int_0^t d\log(M(s)) = \int_0^t -b(s)ds - \int_0^t \sigma dB(s), \\ &\log M(t) - \log M(0) = \int_0^t b(s)ds - \sigma B(t), \\ &\frac{M(t)}{M(0)} = e^{[\int_0^t -b(s)ds - \sigma B(t)]}, \\ &M(t) = a \cdot e^{[\int_0^t -b(s)ds - \sigma B(t)]}. \end{aligned} \tag{A.6}$$

And because $M(t) = a - N(t)$, we can get:

$$\begin{aligned} N(t) &= a - M(t) \\ &= a - a \cdot e^{[\int_0^t -b(s)ds - \sigma B(t)]} \\ &= a(1 - e^{[\int_0^t -b(s)ds - \sigma B(t)]}). \end{aligned} \tag{A.7}$$

In particular, when $b(s) = b$, there is

$$N(t) = a[1 - e^{(-bt - \sigma B(t))}],$$
$$E[e^{\sigma B(t)}] = e^{\frac{1}{2}\sigma^2 t}.$$

Thus,

$$m(t) = E[N(t)] = a \cdot [1 - e^{(-bt - \frac{1}{2}\sigma^2 t)}].$$

Acknowledgement

This work was supported by National Natural Science Foundation of China (No.71901078, 72161005), Science and Technology Foundation of Guizhou (No. QianKeHeJiChu-ZK[2022]-General184), Guizhou Internet + Industrial Technology Research Institute and Key Laboratory of Electric Power Big Data of Guizhou Province.

Reference

1. Cai J, Shi Y, Liu B. Inference for A Series System with Dependent Masked Data Under Progressive Interval Censoring. Journal of Applied Statistics, 2017, 44(1): 3-15. https://doi.org/10.1080/02664763.2016.1156658.
2. Calin O. An Informal Introduction to Stochastic Calculus with Applications. World Scientific, 2015: 117-138. https://doi.org/10.1142/9789814678940_0005.
3. Chatterjee S, Chaudhuri B, Bhar C. Optimal Release Time Determination in Intuitionistic Fuzzy Environment Involving Randomized Cost Budget for SDE-based Software Reliability Growth Model. Arabian Journal for Science and Engineering, 2020, 45(4): 2721-2741. https://doi.org/10.1007/s13369-019-04128-7.
4. Chatterjee S, Chaudhuri B, Bhar C. Optimal Release Time Determination Via Fuzzy Goal Programming Approach for SDE-based Software Reliability Growth Model. Soft Computing, 2021, 25(5): 3545-3564. https://doi.org/10.1007/s00500-020-05385-7.
5. Dohi T, Zheng J, Okamura H. Data-driven Software Reliability Evaluation under Incomplete Knowledge on Fault Count Distribution. Quality Engineering, 2020, 32(3): 421-433. https://doi.org/10.1080/08982112.2020.1757705.
6. Erto P, Giorgio M, Lepore A. The Generalized Inflection S-shaped Software Reliability Growth Model. IEEE Transactions on Reliability, 2018, 69(1): 228-244. https://doi.org/10.1109/TR. 2018. 2869466.
7. HE M L, et al. Additive Stochastic Differential Equations Software Reliability Models. Advances in Applied Mathematics, 2022, 11: 6401. https://doi. org/10. 12677/AMM. 2022.119677. (In Chinese)

8. Huang Y S, Chiu K C, Chen W M. A Software Reliability Growth Model for Imperfect Debugging. Journal of Systems and Software, 2022, 188: 111267. https://doi. org/10. 1016/j. jss. 2022. 111267.
9. Kelley C T. Iterative Methods for Linear and Nonlinear Equations. Society for Industrial and Applied Mathematics, 1995. ISBN: 0-89871-352-8. https://doi.org/10.1137/1.9781611970944
10. Kyurkchiev N, Markov S. On the Hausdorff Distance between the Heaviside Step Function and Verhulst Logistic Function. Journal of Mathematical Chemistry, 2016, 54(1): 109-119. https://doi.org/10.1007/s10910-015-0552-0.
11. Lang W, Deng S, Shu X B, et al. Existence and Ulam-Hyers-Rassias Stability of Stochastic Differential Equations with Random Impulses. Filomat, 2021, 35(2): 399-407. https://doi. org/10. 2298/FIL2102399L.
12. Levy B C. Wiener Process and White Gaussian Noise. Random Processes with Applications to Circuits and Communications. Springer, Cham, 2020: 207-234.https://doi.org/10.1007/978-3-030-22297-0-6.
13. Li L. Software Reliability Growth Fault Correction Model Based on Machine Learning and Neural Network Algorithm. Microprocessors and Microsystems, 2021, 80: 103538. https://doi.org/10.1016/j.micpro.2020.103538.
14. Liu Z, Yang S, Yang M, et al. Software Belief Reliability Growth Model Based on Uncertain Differential Equation. IEEE Transactions on Reliability, 2022. https://doi. org/10. 1109/TR. 2022.3154770.
15. Nguyen H C, Huynh Q T. New Non-Homogeneous Poisson Process Software Reliability Model Based on A 3-arameter S-shaped Function. IET Software, 2022, 16(2): 214-232. https://doi.org/10.1049/sfw2.12055.
16. Pavlov, N., et al. Some Software Reliability Models: Approximation and Modeling Aspects, LAP LAMBERT Academic Publishing, 2018, ISBN: 978-613-9-82805-0.
17. Pavlov N, Iliev A, Rahnev A, et al. Nonstandard Models in Debugging Theory (Part 2). LAP LAMBERT Academic Publishing, 2018. ISBN: 978-613-9-87794-2.
18. Pavlov N, Iliev A, Rahnev A, et al. Analysis of the Chen's and Pham's Software Reliability Models. Cybernetics and Information Technologies, 2018, 18(3): 37-47. https://doi.org/10.2478/cait-2018-0037.
19. Saraf I, Iqbal J, Shrivastava A K, et al. Modelling Reliability Growth for Multi-version Open Source Software Considering Varied Testing and Debugging Factors. Quality and Reliability Engineering International, 2022, 38(4): 1814-1825. https://doi.org/10.1002/pre.3048.
20. Shen L, Sun J T. Existence and Uniqueness of Solutions for Stochastic Impulsive Differential Equations. Stochastics and Dynamics, 2010, 10(03): 375-383. https://doi.org/10. 1142/S0219493710003005.
21. Shen L J, Sun J T. Global Existence of Solutions for Stochastic Impulsive Differential Equations. Acta Mathematica Sinica, English Series, 2011, 27(4): 773-780. https://doi. org/10. 1007/s10114-011-8650-9.
22. Sudharson D, Prabha D. A Novel Machine Learning Approach for Software Reliability Growth Modelling with Pareto Distribution Function. Soft Computing, 2019, 23(18): 8379-8387. https://doi.org/10.1007/s00500-019-04047-7.
23. Tamura Y, Yamada S. Optimisation Analysis for Reliability Assessment Based on Stochastic Differential Equation Modelling for Open Source Software. International Journal of Systems Science, 2009, 40(4): 429-438. https: //doi. org/10. 1080/00207720802556245.
24. Wang J, Wu Z. Study of The Nonlinear Imperfect Software Debugging Model. Reliability Engineering & System Safety, 2016, 153: 180-192. https://doi.org/10.1016/j.ress.2016.05.003.
25. Wang J. Open Source Software Reliability Model with Nonlinear Fault Detection and Fault Introduction. Journal of Software: Evolution and Process, 2021, 33(12): e2385. https://doi.org/10.1002/smr.2385.
26. Yamada S, Kimura M, Tanaka H, et al. Software Reliability Measurement and Assessment with Stochastic Differential Equations. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 1994, 77(1): 109-116. https://cir.nii.ac.jp/crid/1574231874658900992.
27. Yamada S, Tamura Y. OSS Reliability Measurement and Assessment. Switzerland: Springer International Publishing, 2016. https://doi.org/10.1007/978-3-319-31818-9.
28. Yang J F, Zhao M, Hu W. Web Software Reliability Modeling with Random Impulsive Shocks. Journal of Systems Engineering and Electronics, 2014, 25(2): 349-356. https: //doi.org/10. 1109/JSEE. 2014.00040.
29. Zhang C, et al. Survey of Software Reliability Growth Model. Journal of Software, 2017, 28(9): 2402-2430. https://doi.org/10.13328/j.cnki.jos.005306. (In Chinese)

30. Zhang Y, Zhao M, Zhang S, et al. An Integrated Approach to Estimate Storage Reliability with Initial Failures Based on E-Bayesian Estimates. Reliability Engineering & System Safety, 2017, 159: 24-36. https://doi.org/10.1016/j.ress.2016.10.024.