Research Article

Software Reliability Growth Model with Partial Differential Equation for Various Debugging Processes

Jiajun Xu and Shuzhen Yao

School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Correspondence should be addressed to Jiajun Xu; xujiajun@buaa.edu.cn

Received 19 September 2015; Revised 8 December 2015; Accepted 20 December 2015

Academic Editor: Jean-Christophe Ponsart

Copyright © 2016 J. Xu and S. Yao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most Software Reliability Growth Models (SRGMs) based on the Nonhomogeneous Poisson Process (NHPP) generally assume perfect or imperfect debugging. However, environmental factors introduce great uncertainty for SRGMs in the development and testing phase. We propose a novel NHPP model based on partial differential equation (PDE), to quantify the uncertainties associated with perfect or imperfect debugging process. We represent the environmental uncertainties collectively as a noise of arbitrary correlation. Under the new stochastic framework, one could compute the full statistical information of the debugging process, for example, its probabilistic density function (PDF). Through a number of comparisons with historical data and existing methods, such as the classic NHPP model, the proposed model exhibits a closer fitting to observation. In addition to conventional focus on the mean value of fault detection, the newly derived full statistical information could further help software developers make decisions on system maintenance and risk assessment.

1. Introduction

Software reliability, defined as the probability of failure-free operation under certain conditions and by specific time [1], is one of the significant attributes of software systems development life cycle. As the software systems mature with ever growing complexity, evaluation, prediction, and improvement of their reliability become a crucial and daunting task for developers in both the development and testing phases. Numerous Software Reliability Growth Models (SRGMs) have been developed [2–5], which generally agree that the fault debugging process is a Nonhomogeneous Poisson Process (NHPP) under various diverging assumptions: perfect and imperfect debugging phenomenon [6, 7].

A common assumption is that removing detected faults will not introduce new faults, usually called perfect debugging phenomenon. Based on earlier works of Jelinski and Moranda [8], Goel and Okumoto [9] developed the exponential Software Reliability Growth Models (G-O model) with a constant fault detection rate. In latter models, the smoothly changeable fault detection rates are incorporated. Yamada et al. [10] found the fault detection rate fit to a S-shape growth curve and proposed the delay S-shaped SRGM. Employing the learning phenomenon in software fault detection process, Ohba [11] later developed an alternative inflection S-shaped SRGM. Further work was conducted by Yamada and others [12] by incorporating the relationship between the work effort and the number of faults detected into the model. Huang et al. [13] stated that fault detection and repair process are different in software development and operation. They proposed a unified theory to merge multiple moves into a NHPP-based SRGM and found a regularly changeable fault detection rate in the detection process. Later on Huang and Lyu [6] also used multiple change points to deal with imperfect debugging problems. The common assumption is that the removal of the detected faults will not introduce new faults. This scenario is known as perfect debugging. However, given the complexity of the software debugging process, debugging activity may be imperfect in practical software development environment and thus those perfect models may oversimplify the underlying dynamics.

To solve such problems, imperfect debugging processes are incorporated in latter models. Whenever the software system is developed, new errors can be introduced into the
software during development/testing phase and faults may not be corrected perfectly. This phenomenon is known as imperfect debugging. According to the above two phenomena, SRGMs can be divided into two categories: perfect and imperfect debugging model. Kapur and Garg [14] discussed their fault removal phenomenon that as the team gains more experience, they may remove detected faults without causing new ones. However, the testing team may not correct a fault perfectly and the original fault may remain or generate new faults, leading to a phenomenon known as imperfect debugging. It was Goel [15] who first introduced the concept of imperfect debugging. Latter, Ohba and Chou [16] developed an error generation model based on the G-O model, named as an imperfect debugging model. Pham et al. [17] proposed a general imperfect software debugging model (P-N-Z model). Pham [18] also developed an SRGM for multiple failure types incorporating error generation. Zhang et al. [19] proposed a testing efficiency model that includes both imperfect debugging and error generation. Kapur et al. [20] proposed a flexible SRGM with imperfect debugging and error generation using a logistic function for the fault detection rate, which reflects the efficiency of the testing and removal team. Employing the learning phenomenon in software fault detection process, Yamada et al. [21] later developed an imperfect debugging model. In imperfect software debugging models, the fault introduction rate per fault is generally assumed to be constant or decreasing over time [22]. Recently, Wang et al. [22] also proposed a model to represent the imperfect debugging process with a logistic distribution. To sum up, most models presume a deterministic relationship between the cumulative number of detected faults and the time span of the software fault debugging process.

However, software debugging is a stochastic and uncertain process, which can be affected by many environmental factors, such as the distribution of resources, strategy, and running environment [23]. In particular, in the uncertain network environment, the aforementioned assumptions of deterministic model would become problematic. The environmental noise introduces great uncertainties that significantly affects traditional debugging process. To address such challenges, new stochastic models [24–27] were proposed, which treat the debugging process as perfect and stochastic; they assume each failure occurrence is independent and follows identical random distribution. By employing the white noise, that is, temporally uncorrelated random variables, for the environmental factors collectively, they used a flexible stochastic differential equation to model the irregular changes. Comparing to conventional models, such white-noise-based approach assumes the perfect debugging and no doubt provides closer approximation to uncertain fluctuations in reality but with great mathematical simplicity. Debugging activity is usually imperfect in piratical software development and recent data [28, 29] show that the fault detection is highly susceptible to noise and is generally correlated; thus earlier assumptions become problematic. Thus, because of its mathematical simplicity, it may also considerably underestimate the imperfect debugging process and the temporal correlation in a dynamic environment.

In this study, we propose an alternative stochastic framework based on partial differential equation (PDE), to describe the perfect/imperfect debugging processes, in which the environmental noise is of arbitrary correlation. Details of the model is provided in Section 2 with an equation for the probabilistic density function (PDF) of system states. In Section 3, we validate our approach with both historical failure data and results from conventional methods; main features of the model, including the temporal evolutions of its full statistical information, are later addressed in Section 4. The final conclusions are given in Section 5.

2. Problem Formulation

2.1. Model Formulation. Widely used in many practical applications, NHPP model assumes the following:

1. the fault debugging process is modeled as a stochastic process within a continuous state space;
2. the system is subject to failures at random times caused by the remaining errors in the system;
3. the mean number of detected faults in a time interval is proportional to the mean number of remaining faults;
4. when a fault is detected, it may be removed with the probability p;
5. when a fault is removed, new faults may be generated with a constant probability q.

In order to describe the stochastic and uncertain behavior of the fault debugging process, in a continuous state space, the temporal evolution of $M(t)$ is routinely described as [12]

\[
\frac{dM}{dt} = r(t)(a + qM - pM),
\]

where $M(t)$ is the expected number of faults detected in the time interval $[0,t]$, $a$ denotes the number of total faults, and $r(t)$ is a nonnegative function representing the fault detection rate per remaining fault at testing time $t$. We further note that $q(t)M$ is the number of newly introduced bugs, while $p(t)M$ means the number of successful removal; together they represent a (perfect/imperfect) debugging process in which probabilities $q$ and $p$ are assigned [30–32]. Without a loss of generality, we denote $k(t) = p(t) - q(t)$ to facilitate subsequent presentation and $k = 1$ indicates a perfect debugging process.

Since $r(t)$ is subject to a number of random environmental effects [23], we follow the previous studies [24, 25, 27] and represent $r(t)$ as a sum of its mean value $\langle r(t) \rangle$ and a zero-mean (random) fluctuation $r' (t)$:

\[
r(t) = \langle r(t) \rangle + r'(t), \quad \langle r'(t) \rangle = 0.
\]

By substituting (2) into (1), we obtain a stochastic differential equation:

\[
\frac{dM}{dt} = \left[ \langle r(t) \rangle + r'(t) \right] [a - k(t)M].
\]
In contrast to earlier works of [24, 25, 27] which approximate the fluctuations term \( r'(t) \) as a white noise, we relax such assumption and denote the noise's two-point covariance more broadly as

\[
\langle r'(t) r'(s) \rangle = \sigma_r^2 \rho \left( \frac{t-s}{\tau} \right),
\]

where \( \sigma_r^2 \) represents the strength of the noise, \( \rho \) denotes its correlation function, \( t \) and \( s \) indicate two temporal points, and \( \tau \) is the noise correlation time.

2.2. PDF Method. Our goal is to derive an equation for the probabilistic density function of \( M_t \), that is, \( f_M(m; t) \). To be specific, we adopt the PDF method originally developed in the context of turbulent flow [33] and use the Dirac delta function \( \delta(\cdot) \) to define a "raw" PDF of system states \( M \) at time \( t \):

\[
\Pi(m, M; t) = \delta [M(t) - m],
\]

whose ensemble average over random realisations of \( M \) yields \( f_M(m; t) \):

\[
\langle \Pi \rangle \equiv \int \delta (M' - m) f_M (M'; t) \, dM' = f_M (m; t). \tag{6}
\]

Following recent study on PDF method [34], we multiply the original equation (3) with \( \partial \Pi / \partial m \) and apply the properties of the Dirac delta function \( \delta(m - M)g(M) = \delta(m - M)g(m) \). This would yield the stochastic evolution equation for \( \Pi \) in the probability space (\( m \in \Omega \)):

\[
\frac{\partial \Pi}{\partial t} = - \frac{\partial}{\partial m} \left[ \left\langle r(t) \right\rangle + r'(t) \right] [a - k(t) m] \Pi. \tag{7}
\]

Now we take the ensemble average of (7) and apply Definition (6); by employing a closure approximation, variously known as the large-eddy diffusivity (LED) closure [35] or the weak approximation [36], an equation of \( f_M \) can be obtained

\[
\frac{\partial f_M}{\partial t} = - \frac{\partial}{\partial m} (\mathcal{V} f_M) + \frac{\partial}{\partial m} \left( \frac{\partial f_M}{\partial m} \right), \tag{8a}
\]

where the eddy diffusivity \( \mathcal{D} \) and effective velocity \( \mathcal{V} \) are

\[
\mathcal{D} (m, t) = \int_0^t \int_\Omega \langle r'(t) r'(s) \rangle [a - k(t) m] [a - k(s) y] \, \mathcal{G}_d \, dy \, ds,
\]

\[
\mathcal{V} (m, t) = [a - k(t) m] \left\langle r(t) \right\rangle + \int_0^t \int_\Omega \langle r'(t) r'(s) \rangle [a - k(t) m] k(s) \mathcal{G}_d \, dy \, ds. \tag{8b}
\]

And Green’s function \( \mathcal{G}_d(y; s; m; t) \) satisfies the deterministic partial differential equation

\[
\frac{\partial \mathcal{G}_d}{\partial s} + [a - k(s) y] \left\langle r(s) \right\rangle \frac{\partial \mathcal{G}_d}{\partial y} = - \delta \left( y - m \right) \delta (s - t), \tag{8d}
\]

subject to the homogeneous initial (and boundary) conditions corresponding to their (possibly inhomogeneous) counterparts for the raw PDF (7).

3. Model Validation

In this section, we validate our PDE model (8a), (8b), (8c), and (8d) by computing the mean fault detection \( \langle M \rangle \) and compare it with historical data and two other methods, namely, the classic deterministic SRGMs (generalised NHPP model) and the popular stochastic SRGMs (white-noise model). As shown in Table 1, four sets of data are used: the first one (DS1) is from WebERP system [37]; the second (DS2) originated from the open source project management software [38]; and the third (DS3) and fourth (DS4) are from the first and fourth product releases of the software products at Tandem Computers [32]. Among those four sets of data, the two UDP/TCP-based systems (DS1 and DS2) are more likely to be affected by environmental factors, whereas the TCP-based software development at Tandem Computers Inc. takes place in a more enclosed/stable environment. UDP is a connectionless-oriented transport protocol and will introduce larger correlation strength for system based on it, while TCP is connection-oriented and will generate smaller correlation strength. We encapsulate such difference in terms of noise variance, for example, larger \( \sigma_r^2 \) for UDP/TCP-based systems (DS1 and DS2). Moreover, DS1, DS2, and DS3 are from the imperfect debugging process, while DS4 is from the perfect debugging process; we demonstrate this diverse with various configurations of the subsequent presentation \( k \). The testing time of DS1, DS2, DS3, and DS4 is 60 (months), 29 (weeks), 20 (weeks), and 19 (weeks), respectively. For detailed information, please refer to the data values tabulated in the Appendix.

It is also noted here that we consider a special case, that is, constant \( k(t) = k \), by adopting earlier works [30, 39, 40]. A stationary Gaussian process is prescribed to the fluctuation term, \( r' \sim \mathcal{N}(0, \sigma_r^2) \), with an exponential correlation \( \rho(t, s; \tau) = \exp(-|t - s|/\tau) \).

3.1. Model Solution for Constant \( k \). For model validation, we adopt the popular generalized Goel-Okumoto model (G-O), as this enables us to derive an analytical solution of (3) using separation of variables. The expression for the mean fault detection rate per remaining fault using the generalised G-O model is given by [41]

\[
\left\langle r(t) \right\rangle = bt^d, \tag{9}
\]

where \( b \in [0, 1] \) is a constant and \( d \) is the shape factor.

Following the PDF method, we can obtain an analytical solution of \( f_M \):

\[
f_M (m; t) = \left\{ \begin{array}{ll}
\frac{\exp \left[ - (\bar{m} + \bar{r}) \sqrt{2} / \sigma_r^2 \right]}{\sigma_r (a - km) \sqrt{2\pi}}, & m < a \\
0, & m \geq a,
\end{array} \right. \tag{10a}
\]
Table 1: Data sets for model validation.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Time unit</th>
<th>Detected failure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1 [37, 38]</td>
<td>60 months</td>
<td>146</td>
<td>Web-based integrated accounting ERP system (WebERP) from August 2003 to July 2008 which uses HTTP on top of UDP/TCP</td>
</tr>
<tr>
<td>DS2 [37, 38]</td>
<td>49 weeks</td>
<td>94</td>
<td>Open source project management software (OpenProj) from August 2007 to July 2008 used with Remote Display Protocol (RDP) over UDP/TCP</td>
</tr>
<tr>
<td>DS3 [32]</td>
<td>20 weeks</td>
<td>100</td>
<td>Release 1 of software testing data collected from Tandem Computers which uses a simple Session Control Protocol (SCP) on top of TCP</td>
</tr>
<tr>
<td>DS4 [32]</td>
<td>19 weeks</td>
<td>42</td>
<td>Release 4 of software testing data collected from Tandem Computers applying (SCP) on top of TCP</td>
</tr>
</tbody>
</table>

Table 2: Approximation schemes [51] of PDF for the random integral: \( I_t = \int_0^t r(t') \, dt' = \sum_{i=1}^N \int_{(i-1)\Delta}^{i\Delta} r(t') \, dt' \), where \( \Delta = t/N \).

<table>
<thead>
<tr>
<th>Cases</th>
<th>( t \ll \tau )</th>
<th>( t \gg \tau )</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance ( \sigma_t^2 )</td>
<td>( \sigma_t'^2 )</td>
<td>( 2\sigma_t'^2 )</td>
<td>( 2\sigma_t'^2 \left[ \int_0^\Delta (\Delta - t') \rho(t') , dt' + \sum_{i=1}^N \int_{(i-1)\Delta}^{i\Delta} \rho(t' - t'') , dt' , dt'' \right] )</td>
</tr>
<tr>
<td>( \frac{d\sigma_t'}{dt(T)} )</td>
<td>( \sigma_r )</td>
<td>( \frac{1}{\sqrt{2(\Delta + t)}} )</td>
<td>( \frac{1}{\sigma_t} \left[ \int_0^\Delta \rho(t') , dt' + \sum_{i=1}^N \int_{(i-1)\Delta}^{i\Delta} \rho(i\Delta - t'') , dt'' - \int_0^\Delta \rho(t') , dt' + \sum_{i=1}^N \int_{(i-1)\Delta}^{i\Delta} \rho(t' - \Delta) , dt' \right] )</td>
</tr>
</tbody>
</table>

where
\[
\bar{m} = \frac{\ln(1 - km/a)}{k},
\]
\[
\bar{t} = \frac{bt^{d+1}}{d + 1},
\]
and \( \sigma_t^2 \) is the variance of the random integral \( I_t = \int_0^t r(t') \, ds \), whose PDF follows a Gaussian distribution, \( f_{I_t} \sim \mathcal{N}(0, \sigma_t^2) \), as outlined in Table 2.

The cumulative density function (CDF) \( F_M(m; t) \) can be derived as \( F_M(m; t) = \int f_M(m'; t) \, dm' \):
\[
F_M(m; t) = \begin{cases} 
\frac{1}{2} \left[ 1 - \text{erf} \left( \frac{\bar{m} + \bar{t}}{\sigma_t \sqrt{2}} \right) \right], & m < a \\
1, & m \geq a,
\end{cases}
\]
and \( \text{erf}(\ast) \in [-1, 1] \) represents the error function.

The mean fault detection \( \langle M \rangle \) is then
\[
\langle M \rangle = \frac{a}{k} \int_{-\infty}^{\infty} \left[ 1 - \exp \left[ -k \left( I_t + \frac{bt^{d+1}}{d + 1} \right) \right] \right] f_{I_t} \, dl_{I_t}.
\]

It is noted here that the fault detection rate model (12) describes an imperfect debugging process for constant \( k \). In many cases, time delay, failed removal, and new faults would lead to the problem that \( k(t) \) is a time-dependent variable [6, 7, 13, 42], so that we are not able to get an analytical solution \( \langle M \rangle \) and can only obtain a numerical solution. In the current study, our focus lies on the uncertainty quantification framework of environmental factors in imperfect debugging process; without a loss of generality, subsequent approach could also be extended to the imperfect debugging process with time-dependent \( k(t) \).

3.2 Performance Criteria. To evaluate the performance of our PDE model, we employ four comparison criteria, for example, the predicted relative error (PRE), mean square error (MSE), Coefficient of Determination \( (R^2) \), and root mean square prediction error (RMSPE).

3.2.1 Predicted Relative Error (PRE). The relative difference between observed and estimated number of faults at time \( t \), known as predictive validity [7, 43], is as follows:
\[
\text{PRE}(t) = \frac{M(t) - N(t)}{N(t)},
\]
where \( N \) is the fault number from observed data at time \( t \). Hence, for a better fit to the data, we seek \( \text{PRE}(t) \to 0 \) [44].

3.2.2 Mean Square Error (MSE). Consider
\[
\text{MSE}(t) = \frac{\sum_{i=1}^K [M(t_i) - N(t_i)]^2}{N_0},
\]
where \( M(t_i) \) and \( N(t_i) \) denote the estimated and observed number of faults detected at time \( t_i \), \( i = 1, 2, \ldots, K \), respectively, \( K \) is the total number of observation data points. A lower value of MSE(t) represents better model prediction [41, 45].

3.2.3 Coefficient of Determination \( (R^2) \). We define this coefficient as the ratio of the sum of squares resulting from the trend model to that from constant model:
\[
R^2(t) = 1 - \frac{\sum_{i=1}^K [M(t_i) - N(t_i)]^2}{\sum_{i=1}^K [N(t_i) - \bar{N}]^2},
\]
where \( \bar{N} \) is the average of detected faults, \( \bar{N} = \sum_{i=1}^K N(t_i)/K \). \( R^2 \in [0, 1] \) measures the percentage of the total variation about the mean accounted for the fitted curve. The larger \( R^2 \) is, the better the model explains the variation in the data.
3.2.4. Root Mean Square Prediction Error (RMSPE). The average of prediction errors (PEs) is known as Bias [46]. The standard deviation of predicted relative variation (PRV) is known as predicted relative variation [46]. The root mean square prediction error (RMSPE) is a measure of the closeness with which the model predicts the observation and is a measure of closeness with which a model predicts the observation [46]:

$$\text{RMSPE}(t) = \sqrt{\text{Bias}^2 + \text{PRV}^2}. \quad (16)$$

Given

$$\text{PE}(t_i) = M(t_i) - N(t_i), \quad (17a)$$

$$\text{Bias} = \sum_{i=1}^{K} \left[ \frac{\text{PE}(t_i)}{K} \right], \quad (17b)$$

$$\text{PRV}(t) = \sqrt{\frac{\sum_{i=1}^{K} (\text{PE}_i - \text{Bias})^2}{K-1}}. \quad (17c)$$

The lower the values of Bias, PRV, and RMSPE, the better the goodness of fit.

3.3. Model Parameter Estimation. There are six input parameters for our PDE method: the total number of initial faults $a$, the fault detection rate $b$, the shape factor $d$, the subsequent presentation $k$, the noise strength $\sigma^2$, and correlation time $\tau$. Given a data set, we used SPSS to estimate the parameters $a$, $b$, and $d$, and the methods for estimating three parameters include method of moments, least square method, maximum likelihood, and Bayesian methods [47–49]. For the subsequent presentation $k$, we use the empirical value of [30, 39, 40]. About the noise parameters, we adopt earlier works on white noise [24] for the strength of our correlated noise $\sigma^2$, which yields the optimal solution as discussed in the sensitivity analysis Section 3.5. Regarding the noise correlation time $\tau$, we take the fact that a software failure is typically repaired within days [50] and hence set $\tau = 0.1$ (month) or (week) for later analysis. A stationary Gaussian process is prescribed to the fluctuation term, $t' \sim N(0, \sigma^2)$, with an exponential correlation $p(t, s; \tau) = \exp(-|t-s|/\tau)$.

3.4. Validation Results. In this section, we validate our PDE model (12) by computing the mean fault detection $\langle M \rangle$ and compare it with historical data and two other methods, namely, the classic deterministic SRGMs (generalised NHPP model) and the popular stochastic SRGMs (white-noise model). Figure 1 shows expected number of faults $\langle M \rangle$ from our PDE model, the G-O model, and the white-noise model against four data sets over the testing time, respectively. Parameters of all three models are adjusted for their best fit. As demonstrated in Figure 1, though our PDE model overestimates the number of faults for DS1 and leads to underestimation for DS2, DS3, and DS4, overall it provides the overall best match to the four data sets, compared with the G-O model and the white-noise model.

Figure 2 shows the PRE results for our PDE model, G-O model, and white-noise model over the testing time. Similar trends are exhibited for all three models: large estimation error (deviation from zero) at the beginning but gradual convergence to observation at later time. This is expected, since few data can be utilised at earlier time to gauge the models’ parameters; as time elapses and more data become available, the models’ accuracy improve and hence their PRE approaches zero. Overall, the PDE model yields the smallest PRE.

We tabulated the aforementioned evaluation results of our PDE model, G-O model, and white-noise model for DS1, DS2, DS3, and DS4 in Table 3. Despite a slightly higher value of PRV for DS1, the PDE model presents best goodness-to-fit with all four data sets in all performance indicators.

3.5. Sensitivity Analysis. Having validated our PDE model with four observation real data sets, we now conduct the sensitivity analysis to study the impact of correlation time $\tau$, strength $\sigma^2$, and function $p(t, s; \tau)$ on our fault prediction.

We first investigate the impact of the correlation time $\tau$. Figure 3 and Table 4 present the PRE, MSE, $R^2$, Bias, PRV, and RMSPE results on DS2 for our PDE model with parameters $a = 156$, $b = 0.5446$, $d = 0.026$, $\sigma^2 = 0.0002228$, and $k = 0.5092$, G-O model with parameters $a = 94$ and $b = 0.146$, and white-noise model with parameters $a = 97$, $b = 0.16246$, $d = 0.026$, $\sigma^2 = 0.0002228$, and $k = 0.5092$. It demonstrates that the PDE model fits best the real data and the repair time is $\tau = 0.1$ week. Hence a correlation time $\tau = 0.1$ (month or week) is a good approximation of fault repair time in reality.

Next we study the impact of correlation strength. Figure 4 and Table 5 show the results with DS2 for our PDE model with parameters $a = 156$, $b = 0.5446$, $d = 0.026$, and $\tau = 0.1$, G-O model with parameters $a = 94$ and $b = 0.146$, and white-noise model with parameters $a = 97$, $b = 0.16246$, $d = 0.026$, and $k = 0.5092$. It demonstrates that $\sigma^2$ has a large effect on the white-noise model, while its influence on the PDE model is very limited.

Lastly we conduct the sensitivity analysis of the correlation function. Five types of correlation function, exponential, bessel, normal, cosine, and cubic, are studied. Figure 5 and Table 6 show the results on DS2 for our PDE model with parameters $a = 156$, $b = 0.5446$, $d = 0.026$, $\sigma^2 = 0.0002228$, $k = 0.5092$, and $\tau = 0.1$, G-O model with parameters $a = 94$ and $b = 0.146$, and white-noise model with parameters $a = 97$, $b = 0.16246$, $d = 0.026$, and $k = 0.5092$. It is clear that the correlation type significantly affect our model estimation. It also demonstrates that the PDE model with exponential correlation function fits best.

4. Reliability Assessment

Having validated our model with historical data, we now apply our framework first to study the detection process behaviour through its full statistical information in Section 4.1 and later to assess software reliability in Section 4.2.

4.1. Temporal Evolution of PDF. In addition to conventional focus on the mean value of fault detection, $\langle M \rangle$, we now
include full statistical information, that is, the probabilistic density function (PDF) and the cumulative density function (CDF), to study system behaviour.

Three snapshots of the PDF $f_M(m; t)$ and CDF $F_M(m; t)$ at time $t = 2$, $10$ and $50$ weeks are plotted in Figure 6, respectively, using $a = 156$, $b = 0.005446$, $d = 0.026$, $\sigma^2 = 0.0002228$, $k = 0.5092$, and $\tau = 0.1$. The uncertainty of our prediction, encapsulated in the width of the PDF and the slope of the CDF, initially rises from $1$ to $10$ weeks; it later falls at $50$ weeks as most of the faults are detected. Here we propose employing the CDF as a criteria to ensure certain levels of reliability. For example, in Figure 6, for a $90\%$ software reliability in $2$, $10$, and $50$ weeks, the minimum number of faults detected must be $3$, $10$, and $44$, respectively.

4.2. Software Reliability and Reliability Growth Rate. Software reliability $R(t; T)$ is broadly defined as the probability that no software fault is detected in the time interval $[T, T + t]$, given that the last fault occurred at time $T$. The software reliability can be expressed as

$$R (t; T) = e^{-\left[(\langle M(T+t)\rangle-\langle M(T)\rangle)\right]},$$

(18)
Thus, the software reliability growth rate, \( RGR(t; T) \), a measure of reliability change rate, can be computed as

\[
RGR(t; T) = \frac{d}{dt} R(t; T)
\]

\[
= a R \int_{-\infty}^{\infty} \left[ \frac{\sigma_t^2 \sigma_I - \sigma_t^2}{\sigma_I^2} (1 - e^{-\beta}) \right. - b (T + t)^d e^{-\beta} f_{1_t} dI_t
\]

where

\[
\beta = k \left( L_t + \frac{b d^{d+1}}{d + 1} \right). \tag{19b}
\]

The temporal derivative of the variance \( d\sigma_t/dt \) is listed in Table 2.

Three snapshots of software reliability, \( R(t; T) \), at time \( T = 2nd, 10th, 50th \) weeks, are presented in Figure 7, using \( a = 156, b = 0.005446, d = 0.026, \sigma_I^2 = 0.0002228, k = 0.5092, \) and \( \tau = 0.1 \). Reliability at all three timeframes exhibits initial drop but such deterioration gradually slows down and \( R(t; T) \) eventually approaches zero as \( t \to \infty \). This is expected,
Table 3: The MSE, $R^2$, Bias PRV, and RMSPE results of PDE model, G-O model, and white-noise model for DS1, DS2, DS3, and DS4.

<table>
<thead>
<tr>
<th>Models</th>
<th>Data set</th>
<th>Setting</th>
<th>MSE</th>
<th>$R^2$</th>
<th>Bias</th>
<th>PRV</th>
<th>RMSPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDE model</td>
<td>DS1</td>
<td>$a = 432, b = 0.04047, d = 0.028, \sigma_r^2 = 0.0002236, k = 0.9536, \text{ and } \tau = 0.1$</td>
<td>419.6179</td>
<td>0.6466</td>
<td>12.5213</td>
<td>16.3490</td>
<td>13.1506</td>
</tr>
<tr>
<td></td>
<td>DS2</td>
<td>$a = 156, b = 0.5446, d = 0.026, \sigma_r^2 = 0.0002228, k = 0.5092, \text{ and } \tau = 0.1$</td>
<td>61.1382</td>
<td>0.8853</td>
<td>-6.9274</td>
<td>3.6637</td>
<td>6.8133</td>
</tr>
<tr>
<td></td>
<td>DS3</td>
<td>$a = 605, b = 0.846, d = 0.025, \sigma_r^2 = 0.0002219, k = 0.9824, \text{ and } \tau = 0.1$</td>
<td>82.1784</td>
<td>0.8989</td>
<td>-6.9221</td>
<td>6.0056</td>
<td>9.1642</td>
</tr>
<tr>
<td></td>
<td>DS4</td>
<td>$a = 269, b = 0.3046, d = 0.024, \sigma_r^2 = 0.0002207, k = 1, \text{ and } \tau = 0.1$</td>
<td>5.9284</td>
<td>0.9682</td>
<td>-0.9743</td>
<td>2.2926</td>
<td>2.4910</td>
</tr>
<tr>
<td>G-O model</td>
<td>DS1</td>
<td>$a = 249 \text{ and } b = 0.0146$</td>
<td>847.9599</td>
<td>0.2858</td>
<td>25.6179</td>
<td>13.9619</td>
<td>29.1755</td>
</tr>
<tr>
<td></td>
<td>DS2</td>
<td>$a = 94 \text{ and } b = 0.146$</td>
<td>495.6554</td>
<td>0.0701</td>
<td>19.3977</td>
<td>11.0395</td>
<td>22.3919</td>
</tr>
<tr>
<td></td>
<td>DS3</td>
<td>$a = 101 \text{ and } b = 0.246$</td>
<td>318.5772</td>
<td>0.6081</td>
<td>13.5753</td>
<td>11.8893</td>
<td>12.7283</td>
</tr>
<tr>
<td></td>
<td>DS4</td>
<td>$a = 173 \text{ and } b = 0.0146$</td>
<td>15.3147</td>
<td>0.9178</td>
<td>-2.7146</td>
<td>2.8961</td>
<td>3.9694</td>
</tr>
<tr>
<td>White-noise model</td>
<td>DS1</td>
<td>$a = 510, b = 0.0126, d = 0.028, k = 0.9536, \text{ and } \sigma_r^2 = 0.0002236$</td>
<td>760.9538</td>
<td>0.3591</td>
<td>23.6059</td>
<td>14.3934</td>
<td>27.6479</td>
</tr>
<tr>
<td></td>
<td>DS2</td>
<td>$a = 97, b = 0.16246, d = 0.026, k = 0.5092, \text{ and } \sigma_r^2 = 0.0002228$</td>
<td>171.8446</td>
<td>0.6776</td>
<td>10.9225</td>
<td>7.3238</td>
<td>13.1506</td>
</tr>
<tr>
<td></td>
<td>DS3</td>
<td>$a = 199, b = 0.226, d = 0.025, k = 0.9824, \text{ and } \sigma_r^2 = 0.0002219$</td>
<td>146.6972</td>
<td>0.8195</td>
<td>8.8329</td>
<td>9.1706</td>
<td>12.7326</td>
</tr>
<tr>
<td></td>
<td>DS4</td>
<td>$a = 377, b = 0.0126, d = 0.024, k = 1.0000, \text{ and } \sigma_r^2 = 0.0002207$</td>
<td>20.4094</td>
<td>0.8905</td>
<td>-3.6981</td>
<td>2.6661</td>
<td>4.5589</td>
</tr>
</tbody>
</table>

Figure 3: Sensitivity analysis of PRE for PDE model, G-O model, and white-noise model with various configurations of $\tau$.

Figure 4: Sensitivity analysis of PRE for PDE model, G-O model, and white-noise model with various configurations of $\sigma_r^2$.

since, at earlier time, most of the faults are yet removed; with elapsing time, more are debugged and this leads to a fall in software reliability; when the system is cleared of all faults, $R(t; T)$ reaches a stationary state. Therefore, later timeframe ($T = 50$th week) also presents a sharper drop to zero.

For a closer examination, we now look at the software reliability growth rate, $\text{RGR}(t; T)$, at $T = 2$nd, 10th, 50th weeks. As shown in Figure 7, RGR in all three timeframes demonstrates an initial rise to maximum and late fall to stationarity of zero, albeit earlier timeframe ($T = 2$nd week) exhibits higher zenith ($\text{RGR} = 600$), since fewer faults are removed at earlier stage.
Table 4: Various configurations of $\tau$ in obtaining MSE, $R^2$, Bias, PRV, and RMSPE values of PDE model, G-O model, and white-noise model.

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE</th>
<th>$R^2$</th>
<th>Bias</th>
<th>PRV</th>
<th>RMSPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-O model</td>
<td>495.6554</td>
<td>0.0701</td>
<td>19.3977</td>
<td>11.0395</td>
<td>22.3191</td>
</tr>
<tr>
<td>White-noise model</td>
<td>171.8446</td>
<td>0.6776</td>
<td>10.9225</td>
<td>7.3238</td>
<td>13.1506</td>
</tr>
<tr>
<td>PDE model with $\tau = 0.05$</td>
<td>126.5667</td>
<td>0.7625</td>
<td>5.1685</td>
<td>10.0962</td>
<td>11.3423</td>
</tr>
<tr>
<td>PDE model with $\tau = 0.10$</td>
<td>61.1382</td>
<td>0.8853</td>
<td>$-6.9274$</td>
<td>3.6637</td>
<td>7.8366</td>
</tr>
<tr>
<td>PDE model with $\tau = 0.15$</td>
<td>146.7881</td>
<td>0.7244</td>
<td>$-11.9221$</td>
<td>2.1791</td>
<td>12.1196</td>
</tr>
<tr>
<td>PDE model with $\tau = 0.20$</td>
<td>222.5672</td>
<td>0.5824</td>
<td>$-14.6230$</td>
<td>2.9862</td>
<td>14.9248</td>
</tr>
<tr>
<td>PDE model with $\tau = 0.25$</td>
<td>280.6167</td>
<td>0.4735</td>
<td>$-16.3072$</td>
<td>3.8729</td>
<td>16.7608</td>
</tr>
</tbody>
</table>

Table 5: Various configurations of $\sigma^2_{\tau}$ in obtaining MSE, $R^2$, Bias, PRV, and RMSPE values of PDE model, G-O model, and white-noise model.

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE</th>
<th>$R^2$</th>
<th>Bias</th>
<th>PRV</th>
<th>RMSPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-O model</td>
<td>495.6554</td>
<td>0.0701</td>
<td>19.3977</td>
<td>11.0395</td>
<td>22.3191</td>
</tr>
<tr>
<td>White noise model with $\sigma^2_{\tau} = 0$</td>
<td>171.6587</td>
<td>0.6779</td>
<td>10.9154</td>
<td>7.3217</td>
<td>13.1435</td>
</tr>
<tr>
<td>White noise model with $\sigma^2_{\tau} = 0.0002228$</td>
<td>171.8446</td>
<td>0.6776</td>
<td>10.9225</td>
<td>7.3238</td>
<td>13.1506</td>
</tr>
<tr>
<td>White noise model with $\sigma^2_{\tau} = 0.2228$</td>
<td>343.6834</td>
<td>0.3552</td>
<td>16.1015</td>
<td>9.2835</td>
<td>18.5861</td>
</tr>
<tr>
<td>PDE model with $\sigma^2_{\tau} = 0$</td>
<td>61.1445</td>
<td>0.8853</td>
<td>$-6.9277$</td>
<td>3.6640</td>
<td>7.8370</td>
</tr>
<tr>
<td>PDE model with $\sigma^2_{\tau} = 0.0002228$</td>
<td>61.1382</td>
<td>0.8853</td>
<td>$-6.9274$</td>
<td>3.6637</td>
<td>7.8366</td>
</tr>
<tr>
<td>PDE model with $\sigma^2_{\tau} = 0.2228$</td>
<td>56.5464</td>
<td>0.8939</td>
<td>$-6.6843$</td>
<td>3.4804</td>
<td>7.5362</td>
</tr>
</tbody>
</table>

Figure 5: Sensitivity analysis of PRE for PDE model, G-O model, and white-noise model with various correlation functions.

5. Conclusion

We presented a novel model to quantify uncertainties associated with the perfect or imperfect debugging process. Treating the random environmental impacts collectively as a noise with arbitrary correlation, an equation of the distribution of fault detection, that is, its probabilistic density function (PDF), is derived. Our analysis leads to the following major conclusions.

(i) Under the conventional evaluation criteria on mean value of detected fault in the perfect or imperfect debugging process, our model provides best fit to observation data comparing to the classic G-O model and white-noise model.

(ii) The proposed approach also goes beyond uncertainty quantification based on mean and variance of system states by computing its PDF and cumulative density function (CDF). This enables one to evaluate probabilities of rare events, which are necessary for probabilistic risk assessment.

(iii) Predictive uncertainty of fault detection initially increases but gradually diminishes as time elapses; hence more maintenance is needed at the initial stage.

(iv) Software developers could also use CDF as a new criteria to assess system reliability, for example, by computing the minimum number of faults detected for an prescribed confidence level.

(v) The proposed correlated model describes an imperfect debugging process for constant $k$, which enables us to obtain an analytical solution $\langle M(t) \rangle$. The approach could also be extended to the imperfect debugging process with time-dependent $k(t)$, to help us drive a numerical solution by computing its PDF. This would be the focus of our future work.
Figure 6: Snapshots of the probabilistic density function (PDF), $f_M(m; t)$, and the cumulative density function (CDF), $F_M(m; t)$, at time $t = 2$, 10 and 50 weeks, using $a = 156$, $b = 0.005446$, $d = 0.026$, $\sigma_r^2 = 0.0002228$, $k = 0.5092$, and $\tau = 0.1$.

Figure 7: Temporal Evolutions of software reliability, $R(t; T)$, and software reliability growth rate, $RGR(t; T)$, at time $T = 2$nd, 10th, 50th week, using $a = 156$, $b = 0.005446$, $d = 0.026$, $\sigma_r^2 = 0.0002228$, $k = 0.5092$, and $\tau = 0.1$.

Table 6: Various correlation functions in obtaining MSE, $R^2$, Bias, PRV, and RMSPE values of PDE model, G-O model, and white-noise model.

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE</th>
<th>$R^2$</th>
<th>Bias</th>
<th>PRV</th>
<th>RMSPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-O model</td>
<td>495.6554</td>
<td>0.0701</td>
<td>19.3977</td>
<td>11.0395</td>
<td>22.3191</td>
</tr>
<tr>
<td>White-noise model</td>
<td>171.8446</td>
<td>0.6776</td>
<td>10.9225</td>
<td>7.3238</td>
<td>13.1506</td>
</tr>
<tr>
<td>PDE model with exponential correlation</td>
<td>61.1382</td>
<td>0.8853</td>
<td>−6.9274</td>
<td>3.6637</td>
<td>7.8366</td>
</tr>
<tr>
<td>PDE model with besselj correlation</td>
<td>246.4608</td>
<td>0.5374</td>
<td>−15.2180</td>
<td>3.8966</td>
<td>15.7089</td>
</tr>
<tr>
<td>PDE model with normal correlation</td>
<td>602.0697</td>
<td>0.1296</td>
<td>−23.1279</td>
<td>8.2806</td>
<td>24.5656</td>
</tr>
<tr>
<td>PDE model with cosine correlation</td>
<td>84.3103</td>
<td>0.8418</td>
<td>0.7364</td>
<td>9.2473</td>
<td>9.2766</td>
</tr>
<tr>
<td>PDE model with cubic correlation</td>
<td>287.8605</td>
<td>0.4599</td>
<td>10.2405</td>
<td>13.6676</td>
<td>17.0784</td>
</tr>
</tbody>
</table>
Table 7: The failure data of web-based and integrated accounting ERP system (DS1) from August 2003 to July 2008 [37, 38].

<table>
<thead>
<tr>
<th>Time unit (month)</th>
<th>Detected faults</th>
<th>Corrected faults</th>
<th>Time unit (month)</th>
<th>Detected faults</th>
<th>Corrected faults</th>
<th>Time unit (month)</th>
<th>Detected faults</th>
<th>Corrected faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>53</td>
<td>50</td>
<td>41</td>
<td>72</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>7</td>
<td>22</td>
<td>53</td>
<td>50</td>
<td>42</td>
<td>74</td>
<td>67</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>7</td>
<td>23</td>
<td>53</td>
<td>50</td>
<td>43</td>
<td>74</td>
<td>67</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>9</td>
<td>24</td>
<td>53</td>
<td>50</td>
<td>44</td>
<td>80</td>
<td>67</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>25</td>
<td>53</td>
<td>53</td>
<td>45</td>
<td>84</td>
<td>69</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>9</td>
<td>26</td>
<td>53</td>
<td>53</td>
<td>46</td>
<td>84</td>
<td>76</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>11</td>
<td>27</td>
<td>53</td>
<td>53</td>
<td>47</td>
<td>84</td>
<td>79</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>15</td>
<td>28</td>
<td>53</td>
<td>53</td>
<td>48</td>
<td>84</td>
<td>82</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>15</td>
<td>29</td>
<td>53</td>
<td>53</td>
<td>49</td>
<td>85</td>
<td>83</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>18</td>
<td>30</td>
<td>53</td>
<td>53</td>
<td>50</td>
<td>86</td>
<td>84</td>
</tr>
<tr>
<td>11</td>
<td>18</td>
<td>18</td>
<td>31</td>
<td>53</td>
<td>53</td>
<td>51</td>
<td>89</td>
<td>87</td>
</tr>
<tr>
<td>12</td>
<td>21</td>
<td>21</td>
<td>32</td>
<td>54</td>
<td>54</td>
<td>52</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>13</td>
<td>22</td>
<td>22</td>
<td>33</td>
<td>56</td>
<td>54</td>
<td>53</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>14</td>
<td>22</td>
<td>22</td>
<td>34</td>
<td>58</td>
<td>56</td>
<td>54</td>
<td>92</td>
<td>91</td>
</tr>
<tr>
<td>15</td>
<td>27</td>
<td>26</td>
<td>35</td>
<td>59</td>
<td>56</td>
<td>55</td>
<td>108</td>
<td>106</td>
</tr>
<tr>
<td>16</td>
<td>30</td>
<td>28</td>
<td>36</td>
<td>60</td>
<td>56</td>
<td>56</td>
<td>120</td>
<td>119</td>
</tr>
<tr>
<td>17</td>
<td>45</td>
<td>39</td>
<td>37</td>
<td>63</td>
<td>56</td>
<td>57</td>
<td>128</td>
<td>126</td>
</tr>
<tr>
<td>18</td>
<td>47</td>
<td>44</td>
<td>38</td>
<td>70</td>
<td>60</td>
<td>58</td>
<td>129</td>
<td>128</td>
</tr>
<tr>
<td>19</td>
<td>49</td>
<td>47</td>
<td>39</td>
<td>71</td>
<td>61</td>
<td>59</td>
<td>139</td>
<td>136</td>
</tr>
<tr>
<td>20</td>
<td>51</td>
<td>47</td>
<td>40</td>
<td>71</td>
<td>64</td>
<td>60</td>
<td>146</td>
<td>136</td>
</tr>
</tbody>
</table>

Table 8: The failure data of open source project management software (DS2) from August 2007 to July 2008 [37, 38].

<table>
<thead>
<tr>
<th>Time unit (week)</th>
<th>Detected faults</th>
<th>Corrected faults</th>
<th>Time unit (week)</th>
<th>Detected faults</th>
<th>Corrected faults</th>
<th>Time unit (week)</th>
<th>Detected faults</th>
<th>Corrected faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>0</td>
<td>18</td>
<td>55</td>
<td>31</td>
<td>35</td>
<td>82</td>
<td>41</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>2</td>
<td>19</td>
<td>57</td>
<td>31</td>
<td>36</td>
<td>83</td>
<td>41</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>11</td>
<td>20</td>
<td>58</td>
<td>31</td>
<td>37</td>
<td>83</td>
<td>44</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>12</td>
<td>21</td>
<td>61</td>
<td>31</td>
<td>38</td>
<td>84</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>28</td>
<td>12</td>
<td>22</td>
<td>61</td>
<td>31</td>
<td>39</td>
<td>84</td>
<td>44</td>
</tr>
<tr>
<td>6</td>
<td>29</td>
<td>13</td>
<td>23</td>
<td>64</td>
<td>33</td>
<td>40</td>
<td>85</td>
<td>44</td>
</tr>
<tr>
<td>7</td>
<td>32</td>
<td>14</td>
<td>24</td>
<td>66</td>
<td>33</td>
<td>41</td>
<td>85</td>
<td>45</td>
</tr>
<tr>
<td>8</td>
<td>36</td>
<td>15</td>
<td>25</td>
<td>67</td>
<td>33</td>
<td>42</td>
<td>87</td>
<td>45</td>
</tr>
<tr>
<td>9</td>
<td>36</td>
<td>20</td>
<td>26</td>
<td>69</td>
<td>33</td>
<td>43</td>
<td>87</td>
<td>46</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>21</td>
<td>27</td>
<td>70</td>
<td>33</td>
<td>44</td>
<td>87</td>
<td>46</td>
</tr>
<tr>
<td>11</td>
<td>41</td>
<td>22</td>
<td>28</td>
<td>73</td>
<td>33</td>
<td>45</td>
<td>89</td>
<td>47</td>
</tr>
<tr>
<td>12</td>
<td>41</td>
<td>22</td>
<td>29</td>
<td>74</td>
<td>33</td>
<td>46</td>
<td>89</td>
<td>62</td>
</tr>
<tr>
<td>13</td>
<td>45</td>
<td>24</td>
<td>30</td>
<td>75</td>
<td>36</td>
<td>47</td>
<td>91</td>
<td>62</td>
</tr>
<tr>
<td>14</td>
<td>47</td>
<td>25</td>
<td>31</td>
<td>75</td>
<td>38</td>
<td>48</td>
<td>91</td>
<td>65</td>
</tr>
<tr>
<td>15</td>
<td>49</td>
<td>30</td>
<td>32</td>
<td>78</td>
<td>38</td>
<td>49</td>
<td>94</td>
<td>65</td>
</tr>
<tr>
<td>16</td>
<td>52</td>
<td>30</td>
<td>33</td>
<td>79</td>
<td>40</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>17</td>
<td>52</td>
<td>31</td>
<td>34</td>
<td>80</td>
<td>41</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Appendix

Data Sets

In the Appendix, four data sets used are presented. The detailed information about two data sets DS1 and DS2 can be referenced for further study at the SourceForge website. Originally, these two data sets had belonged to time-between-failures data, as tabulated in Tables 7 and 8. The other software testing data were collected from two major releases of the software products at Tandem Computers [4], as summarized in Table 9.
Table 9: The failure data of tandem software (DS3 and DS4) from the first and fourth release [32].

<table>
<thead>
<tr>
<th>Time unit (week)</th>
<th>Detected faults</th>
<th>Time unit (week)</th>
<th>Detected faults</th>
<th>Time unit (week)</th>
<th>Detected faults</th>
<th>Time unit (week)</th>
<th>Detected faults</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Release 1</td>
<td></td>
<td>Release 1</td>
<td></td>
<td>Release 4</td>
<td></td>
<td>Release 4</td>
</tr>
</tbody>
</table>

1. 16 11 6 8 49 16 98
2. 24 12 2 8 7 17 99
3. 27 13 8 9 8 18 100
4. 33 14 93 9 27 19 42
5. 41 15 96 10 75 20 100

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References


