mathematics

Article

Software Reliability Modeling Incorporating Fault Detection and Fault Correction Processes with Testing Coverage and Fault Amount Dependency

Qiuying Li [1,2,*] and Hoang Pham [3]

[1] School of Reliability & Systems Engineering, Beihang University, Beijing 100191, China
[2] National Key Laboratory of Science & Technology on Reliability & Environmental Engineering, Beijing 100191, China
[3] Department of Industrial & Systems Engineering, Rutgers University, Piscataway, NJ 08854, USA; hopham@rci.rutgers.edu
* Correspondence: li_qiuying@buaa.edu.cn

Abstract: This paper presents a general testing coverage software reliability modeling framework that covers imperfect debugging and considers not only fault detection processes (FDP) but also fault correction processes (FCP). Numerous software reliability growth models have evaluated the reliability of software over the last few decades, but most of them attached importance to modeling the fault detection process rather than modeling the fault correction process. Previous studies analyzed the time dependency between the fault detection and correction processes and modeled the fault correction process as a delayed detection process with a random or deterministic time delay. We study the quantitative dependency between dual processes from the viewpoint of fault amount dependency instead of time dependency, then propose a generalized modeling framework along with imperfect debugging and testing coverage. New models are derived by adopting different testing coverage functions. We compared the performance of these proposed models with existing models under the context of two kinds of failure data, one of which only includes observations of faults detected, and the other includes not only fault detection but also fault correction data. Different parameter estimation methods and performance comparison criteria are presented according to the characteristics of different kinds of datasets. No matter what kind of data, the comparison results reveal that the proposed models generally give improved descriptive and predictive performance than existing models.

Keywords: fault correction process; fault detection process; testing coverage; time dependency; fault amount dependency; software reliability growth model

check for updates

Citation: Li, Q.; Pham, H. Software Reliability Modeling Incorporating Fault Detection and Fault Correction Processes with Testing Coverage and Fault Amount Dependency. *Mathematics* **2022**, *10*, 60. https://doi.org/10.3390/math10010060

Academic Editor: Tadashi Dohi

Received: 3 December 2021
Accepted: 23 December 2021
Published: 24 December 2021

1. Introduction

Software reliability growth models (SRGMs) based on the nonhomogeneous Poisson process (NHPP) have been provided to describe the software reliability in previous eras [1]. One common assumption of most models is that faults will be instantaneously removed after the failure caused by the faults being observed. However, it does not always stand in real software developing process because of the software's complexity and the tester's limited ability, i.e., it needs nontrivial time and effort not only to report, diagnose, and locate a fault but also to fix and verify a fault instead so that the time can be ignored. Therefore, it is very important to model software reliability based on the perspective of the fault correction process. Recently, great importance has been emphasized on modeling fault correction processes.

It is Schneidewind who first modeled the fault correction process (FCP) along with the fault detection process (FDP) by assuming a constant time delay between the two processes [2]. Then Xie et al. gave an extension study on Schneidewind's idea from an equal time lag to a time-varied lag function [3]. Schneidewind also extended his original model by introducing a random time delay obeying exponential distribution [4]. Huang et al. considered how to combine both fault dependency and a time-dependent debugging time lag into reliability modeling, meanwhile they classified the faults into leading faults and dependent faults [5]. Xie et al. provided a modeling framework of a fault correction process by defining a random time delay model with different distributions, such as the gamma [6], normal, Weibull, chi-square, and Erlang distribution [7]. New statistical distributions continued to be presented in follow-up research, such as the hyper-Erlang distribution [8]. Improved joint likelihood functions for combined FDP and FCP were derived to replace the single likelihood function [9]. Peng et al. integrated testing effort function and imperfect debugging into the dual processes modeling, because they regarded the testing effort as a considerable influencing factor, which affects not only the fault detection rate but also the time to correct a detected fault [10]. Later, they introduced fault dependency into the paired FDP and FCP models instead of neglecting the faults' correlation [11]. Lo et al. summarized a framework which could cover some existing models [12]. Recently, this research was extended to a multi-release, open-source software failure process [13] together with different kinds of failure data, such as masked data [14]. Pachauri et al. implemented delay in fault correction after fault detection based on imperfect correction [15]. Tiwari et al. considered inclusive modeling for investigating the detection and correction of faults under an imperfect debugging scenario where new faults are involved throughout the correction of a hard type of fault [16]. Choudhary et al. evaluated the optimum release and cost by considering that either detection or correction does not provide adequate information and proposed an effort-based optimized decision model taking into account the cost of detection and correction separately using a multi-attribute utility theory [17]. Saraf et al. presented a model of two-stage fault detection and fault correction in view of imperfect debugging, error generation, and change point, where a combination of exponential and gamma distribution is adopted [18]. Kumar et al. allocated the resources in an optimal manner to minimize the cost during the testing phase using FDP and FCP under a dynamic environment [19].

Although models proposed by the above studies can effectively evaluate software reliability, they have the following two disadvantages: firstly, time-delay functions are developed by assuming how the correction time lag will be, which may not be the case. Since time-delay functions are not derived from actual testing process, it may not be very realistic to use them to characterize the relationship between fault detection and correction processes. Secondly, stochastic distribution of fault correction time brings more difficulties in modeling and corresponding parameter estimation. Sometimes the calculations may not yield any solutions. Moreover, in essence, they can be attributed to one category, models based on the time dependency between these two processes, i.e., time delay. However, we know that the time to remove faults depends on many factors, such as the complexity of detected faults, the skills of testers, available resources, and the software development environment. Therefore, it is fairly important to model fault detection and correction processes with different software reliability models. To overcome this problem, the objective of this paper is to provide a general framework for the combined modeling of software detection and correction process from another perspective, which is different from the above-mentioned one.

Shu et al. proposed modeling both fault detection and correction processes from the viewpoint of the fault number concretely and in two ways, the ratio of corrected fault number to detected fault number and the difference between these two numbers [20]. Compared to the time dependency models, they opened a new direction for modeling dual processes, but they only presented models under the condition of perfect debugging, and the fault detection rate follows a constant. Thus, a lot of questions affecting the model accuracy remained unsolved, such as imperfect debugging and a more complicated fault detection rate. In this paper, we aim to give a general framework from the viewpoint of fault amount dependency, so that further studies can be motivated to utilize this framework along this way. In addition, we want to provide an extensive discussion on essential factors, which may have a great effect on model accuracy. Recently, we provided a model on this direction, but the limitation of this article is also that only one testing coverage function is provided, and one single process parameter estimation method is discussed [21].

In addition, with the evolution of SRGMs, many factors have been incorporated into the modeling framework to improve the evaluation accuracy [22]. Testing coverage is a very promising metric during software development for both developers and users, which can help developers assess testing progress and help users estimate confidence in accepting software products [23]. Many time-dependent testing coverage functions (TCFs) following different distributions have been proposed [24], such as the logarithmic–exponential [25], S-shaped [26], Rayleigh [27], Weibull & Logistic [28], and lognormal [29], and many TCF-based SRGMs have been proposed, such as the Rayleigh, logarithmic–exponential, Beta, and Hyper-exponential model. Recently, Chatterjee et al. presented a unified approach to model the reliability growth of software with imperfect debugging and three types of testing coverage curves, such as the exponential, Weibull, and S-shaped [30]. They also proposed a SRGM considering the effects of uncertain testing environment and testing coverage of multi-release software in the presence of two different types of faults [31]. Obviously, incorporating coverage in the SRGM helps to enhance software reliability and predict the faults in a more realistic and accurate way.

Furthermore, different parametric SRGMs are developed depending on failure data gathered during software testing. There are two kinds of failure data, one of which includes only detected faults observations, whereas the other includes not only detected fault number but also the corrected fault number. So, different parameter estimation methods have been recommended aiming at these two conditions. Improved joint likelihood functions for the combined FDP and FCP were derived to replace the single likelihood function [9], meanwhile a combined least square error function is presented to replace a single least square error function [6]. Different criteria are also given to reflect the characteristics of the dual processes simultaneously, e.g., combined MSE and MRE instead of those criteria only considering the characteristics of single process [7].

Besides, some other attempts have been made to model dual processes simultaneously. One is the so-called parametric model based on the Markov chain rather than on NHPP [32]. Liu et al. extended the Markov model by focusing on the weighted least square estimation method, which emphasizes the influence of later data on the prediction [33]. Conversely, nonparametric models are proposed to model these two processes together, e.g., data-driven models, such as robust recurrent neural networks [34], artificial neural networks models [35], finite and infinite server queuing models [36], and quasi-renewal, time-delay fault removal models [37]. The simulation rate-based method [38] was developed to simulate software failure processes by queuing models [39]. Chatterjee et al. developed a modeling framework to incorporate imperfect debugging and change-point with the Weibull-type fault reduction factor considering fault removal as a two-step process [40].

The main contributions of our work are as follows.

- We develop a general framework for modeling both fault detection and correction processes from the viewpoint of fault amount dependency instead of time dependency in the context of different testing coverage and imperfect debugging.
- We consider testing coverage functions including the Weibull-type, delayed S-shaped, and inflection S-shaped functions to verify their flexibility in modeling different failure phenomena.
- We discuss the models under two kinds of failure datasets followed by two kinds of parameter estimation methods and performance comparison criteria accordingly.
- We conduct case studies based on two kinds of failure datasets to verify the feasibility of the proposed method.

The remainder of this paper is structured as follows. In Section 2, we analyze the relationship between fault detection and fault correction processes on the basis of one real dataset, then propose three testing coverage functions to build the proposed models.

According to different kinds of failure datasets, we give parameter estimation methods and performance comparison criteria for estimation and prediction accordingly in Section 3. Then, we validate the performance of the proposed models with several existing SRGMs on three real datasets in Section 4. Finally, we summarize the conclusions in Section 5.

2. Model Formulation

2.1. Assumptions

The assumptions are made to develop the proposed models as follows:

1. The software failure process follows an NHPP process.
2. The mean number of faults detected in the time interval $(t + \Delta t)$ is proportional to the number of undetected faults at time t.
3. The fault detection rate is denoted by testing coverage, which is written as $\frac{c'(t)}{1-c(t)}$, where $c(t)$ refers to one kind of testing coverage, e.g., code percentage that has been examined up to time t, and $c'(t)$ is the derivative of $c(t)$.
4. The software debugging process is imperfect, and new faults could be introduced during fault correction.
5. The detected faults cannot be corrected immediately, and the dependency between fault detection and correction processes is represented by $r(t)$, that is, $r(t) = \frac{m_c(t)}{m_d(t)}$, where $m_d(t)$ is the cumulative detected faults, and $m_c(t)$ represents the cumulative corrected faults.

Under the above assumptions, we can obtain the following equations:

$$\frac{\mathrm{d}m_d(t)}{\mathrm{d}t} = \frac{c'(t)}{1-c(t)}[a(t) - m_d(t)] \tag{1}$$

$$a(t) = a + \alpha m_d(t) \tag{2}$$

$$m_c(t) = r(t) \cdot m_d(t) \tag{3}$$

Solving the value of $m_d(t)$ from Equation (1) and using the initial condition that $t = 0$ and $m_d(t) = 0$, we can obtain the paired FDP and FCP models as follows:

$$m_d(t) = \frac{a}{1-\alpha}\left[1 - \left(\frac{1-c(t)}{1-c(0)}\right)^{1-\alpha}\right] \tag{4}$$

$$m_c(t) = \frac{a}{1-\alpha}\left[1 - \left(\frac{1-c(t)}{1-c(0)}\right)^{1-\alpha}\right] \cdot r(t) \tag{5}$$

where $c(0)$ means $c(t = 0)$.

We can obtain different paired mean value functions by substituting different testing coverage functions $c(t)$ and fault amount dependency function $r(t)$ in Equations (4) and (5).

2.2. The Relationship between $m_d(t)$ and $m_c(t)$

Firstly, we study the relationship between the mean number of $m_d(t)$ and $m_c(t)$. A set of failure data collected from testing a real software program (Dataset 1, DS-1) [9] is utilized here. Suppose $r(t) = \frac{m_c(t)}{m_d(t)}$, the points of cumulative detected and corrected faults are plotted in Figure 1a, and the actual ratio values of $r(t)$ are also drawn in Figure 1b.

From Figure 1, it can be noticed that $m_d(t)$ and $m_c(t)$ are both increasing functions of testing time t. At the beginning of software testing, there are many faults, most of which are simple and easy to be detected. Therefore, the number of faults detected grows rapidly, but the testers at this time are not familiar with the software, so they appear very slow in locating and fixing the detected faults, thus the fault correction lags far behind fault detection, that is, the ratio of corrected fault number to detected fault number is very low and even decreases over time. Then, as the testing progresses, under the influence of the learning process, the testers gain more experience and master the software better than before, they can remove the faults faster, so the number of faults corrected grows faster, and the proportion of faults corrected number against faults detected number increases rapidly over time. Then the faults become more complicated and difficult to locate and fix, so the growth of the faults detected number becomes slow, and the difference between FCP and FDP may decrease. At the end of testing, the faults are almost completely detected, and it is more difficult to find more faults, so the number of detected faults becomes harder to grow, and all the faults are almost completely corrected, hence the ratio of faults corrected number against faults detected number tends to 1.

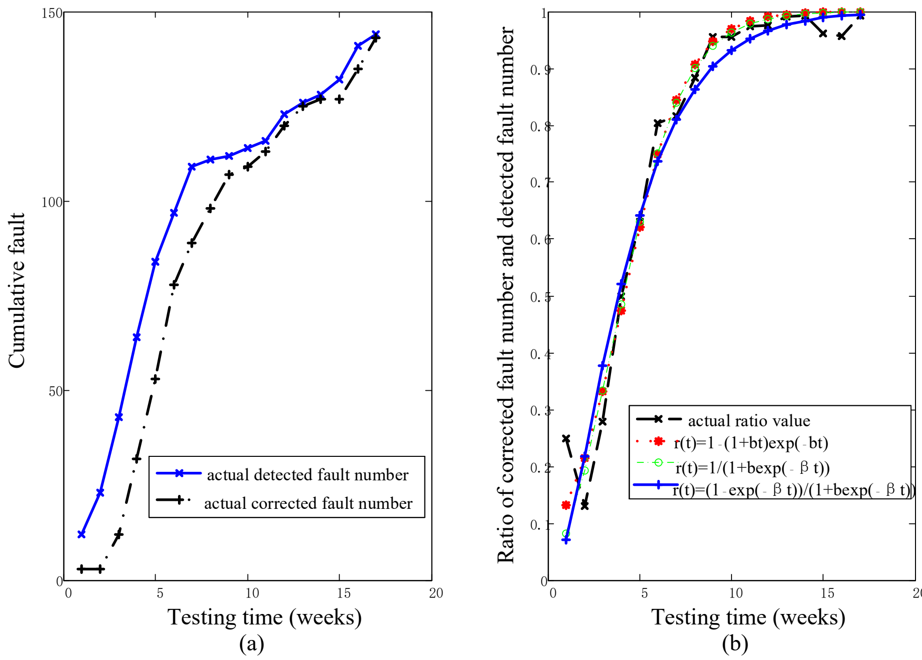

Figure 1. The cumulative number of faults detected and corrected with three function fitting results for the actual ratio on DS-1 (**a**) The actual cumulative number of detected faults and corrected faults for DS-1; (**b**) The actual ratio values and three function fitting results for the actual ratio on DS-1.

Apparently, this phenomenon is better captured by an S-shaped function.

Here three S-shaped functions are used to compare their fitting performance on DS-1, i.e., $r(t) = \frac{1}{1+be^{-\beta t}}$, $r(t) = \frac{1-e^{-\beta t}}{1+be^{-\beta t}}$, and $r(t) = 1 - (1+bt)e^{-bt}$. Table 1 gives their values of MSE, R^2, and Adjusted R^2 on DS-1 separately. From Table 1, $r(t) = \frac{1}{1+be^{-\beta t}}$ yields a better fit for the actual ratio values and is finally decided to be taken as $r(t)$. It can be noted that $r(t) = \frac{1}{1+be^{-\beta t}}$ is a nondecreasing function with an S-shaped curve, whose structure is very flexible and may catch the features of the software testing's learning process.

Table 1. The values of MSE, R^2, and Adjusted R^2 of $r(t)$ for DS-1.

$r(t)$	MSE	R^2	Adjusted R^2
$\frac{1}{1+be^{-\beta t}}$	0.0022	0.9765	0.9749
$\frac{1-e^{-\beta t}}{1+be^{-\beta t}}$	0.0029	0.9695	0.9675
$1-(1+bt)e^{-bt}$	0.0038	0.9567	0.9567

2.3. Framework and New Testing Coverage Models

Here time-variable testing coverage functions are chosen to obtain the following specific models, and these testing coverage functions have been recommended in several references with great flexibility [41].

Model 1: Suppose the testing coverage is a Weibull-type function, that is

$$c(t) = 1 - \mathrm{e}^{-rt^d} \tag{6}$$

We can obtain

$$\begin{cases} m_d(t) = \frac{a}{1-\alpha}\left(1 - \mathrm{e}^{-r(1-\alpha)t^d}\right) \\ m_c(t) = \frac{a}{1-\alpha}\left(1 - \mathrm{e}^{-r(1-\alpha)t^d}\right)\frac{1}{1+b\mathrm{e}^{-\beta t}} \end{cases} \tag{7}$$

where r and d are constant parameters.

Because $\frac{c'(t)}{1-c(t)} = rdt^{d-1}$, the derivative of $\frac{c'(t)}{1-c(t)}$ equals to $rd(d-1)t^{d-1}$. For $d > 1$, the fault detection rate is increasing, and it is decreasing for $d < 1$, and it keeps a constant for $d = 1$. In most actual testing processes, software failure intensity usually increases initially and then decreases, so the flexibility of the Weibull-type function captures the characteristics of the failure intensity.

Model 2: Suppose the testing coverage takes a delayed S-shaped function, that is

$$c(t) = 1 - (1 + rt)\mathrm{e}^{-rt} \tag{8}$$

Then

$$\begin{cases} m_d(t) = \frac{a}{1-\alpha}\left(1 - (1+rt)^{1-\alpha}\mathrm{e}^{-r(1-\alpha)t}\right) \\ m_c(t) = \frac{a}{1-\alpha}\left(1 - (1+rt)^{1-\alpha}\mathrm{e}^{-r(1-\alpha)t}\right) \cdot \frac{1}{1+b\mathrm{e}^{-\beta t}} \end{cases} \tag{9}$$

Because $\frac{c'(t)}{1-c(t)} = \frac{r^2 t}{1+rt}$, the derivative of $\frac{c'(t)}{1-c(t)}$ equals to $\frac{r^2}{(1+rt)^2}$. For any $r > 0$, the fault detection rate keeps increasing.

Model 3: Suppose the testing coverage is an inflection S-shaped function, that is

$$c(t) = \frac{1 - \mathrm{e}^{-rt}}{1 + c\mathrm{e}^{-rt}} \tag{10}$$

then

$$\begin{cases} m_d(t) = \frac{a}{1-\alpha}\left(1 - \left(\frac{(1+c)\mathrm{e}^{-rt}}{1+c\mathrm{e}^{-rt}}\right)^{1-\alpha}\right) \\ m_c(t) = \frac{a}{1-\alpha}\left(1 - \left(\frac{(1+c)\mathrm{e}^{-rt}}{1+c\mathrm{e}^{-rt}}\right)^{1-\alpha}\right) \cdot \frac{1}{1+b\mathrm{e}^{-\beta t}} \end{cases} \tag{11}$$

because $\frac{c'(t)}{1-c(t)} = \frac{r}{1+c\mathrm{e}^{-rt}}$, and the derivative of $\frac{c'(t)}{1-c(t)}$ equals to $\frac{cr^2\mathrm{e}^{-rt}}{(1+c\mathrm{e}^{-rt})^2}$. For any $c > 0, r > 0$, the fault detection rate keeps increasing.

It can be seen that fault detection and correction processes as well testing coverage and imperfect debugging are all integrated into the proposed paired models.

In this paper, two kinds of failure data will be utilized. One kind of failure data contains only the detected fault number, and the other contains not only the detected fault number but also the corrected fault number. DS-I belongs to the second type, and Datasets 2 and 3 belong to the first type, respectively. Table 2 lists 19 models, among which, M1 to M5 are paired FDP and FCP models with a different time delay as comparison models to depict both fault correction and detection processes [6,7]. M6 to M16 are taken as comparison models for single process models. M17 to M19 are proposed models in this paper. Accordingly, for DS-1, the paired models in M17 to M19 will be used to compare the performance, and for DS-2 and DS-3, the FCP models in M17 to M19 will be used to compare the performance.

Table 2. Summary of models.

Models	Model Names	Mean Value Functions	Remarks
M1	FDP: G-O model FCP: G-O with constant time delay [6]	$m_d(t) = a(1 - e^{-bt})$ $m_c(t) = a(1 - e^{-b(t-c)})$	Schneidewind model considers FCP as a constant time delay from FDP. Here taken as comparing models for DS-1.
M2	FDP: G-O model FCP: G-O with time-dependent delay [6]	$m_d(t) = a(1 - e^{-bt})$ $m_c(t) = a(1 - (1 + ct)e^{-bt})$	Xie et al. model considers FCP as a time-dependent delay from FDP. Here taken as comparing models for DS-1.
M3	FDP: G-O model FCP: G-O with exponential distributed delay [6]	$m_d(t) = a(1 - e^{-bt})$ $m_c(t) = a(1 - \frac{c}{c-b}e^{-bt} + \frac{b}{c-b}e^{-ct})$	Assume FCP has an exponential distributed delay from FDP. Taken as comparing models for DS-1.
M4	FDP: G-O model FCP: G-O with normally distributed time delay [7]	$m_d(t) = a(1 - e^{-bt})$ $m_c(t) = -ae^{-bt+\mu b+b^2\sigma^2/2}(\Phi(t, b\sigma^2 + \mu, \sigma) - \Phi(0, b\sigma^2 + \mu, \sigma))$ $+a(\Phi(t, \mu, \sigma) - \Phi(0, \mu, \sigma))$	Assume FCP has a normally distributed time delay from FDP. $f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ Taken as comparing models for DS-1.
M5	FDP: G-O model FCP: G-O with gamma distributed time delay [7]	$m_d(t) = a(1 - e^{-bt})$ $m_c(t) = a\Gamma(t, \alpha, \beta) - \frac{ae^{-bt}}{(1-b\beta)^\alpha}\Gamma(t, \alpha, \frac{\beta}{1-b\beta})$	Assume FCP has a gamma distributed time delay from FDP. $f(x;\alpha,\beta) = x^{\alpha-1}\frac{\beta^\alpha e^{-\beta x}}{\Gamma(\alpha)}, x > 0$ Taken as comparing models for DS-1
M6	G-O model [42]	$m(t) = a(1 - e^{-bt})$	Taken as comparing model for DS-2 and DS-3.
M7	Delayed S-shaped [42]	$m(t) = a(1 - (1 + bt)e^{-bt})$	Taken as comparing model for DS-2 and DS-3.
M8	Inflection S-shaped [43]	$m(t) = \frac{a(1-e^{-bt})}{1+\beta e^{-bt}}$	Taken as comparing model for DS-2 and DS-3.
M9	Yamada exponential [42]	$m(t) = a(1 - e^{-r\alpha(1-e^{-\beta t})})$	Taken as comparing model for DS-2 and DS-3.
M10	Yamada Rayleigh [42]	$m(t) = a(1 - e^{-r\alpha(1-e^{-\beta t^2/2})})$	Taken as comparing model for DS-2 and DS-3.
M11	Yamada Weibull [42]	$m(t) = a(1 - e^{-r\alpha(1-e^{-\beta t^r})})$	Taken as comparing model for DS-2 and DS-3.
M12	Yamada imperfect (1) [44]	$m(t) = \frac{ab}{\alpha+b}(e^{\alpha t} - e^{-bt})$	Taken as comparing model for DS-2 and DS-3.
M13	Yamada imperfect (2) [44]	$m(t) = a(1 - e^{-bt})(1 - \frac{\alpha}{b}) + \alpha at$	Taken as comparing model for DS-2 and DS-3.
M14	P-Z (1997) model [45]	$m(t) = \frac{1}{(1+\beta e^{-bt})}\left((c + a)(1 - e^{-bt}) - \frac{ab}{b-\alpha}(e^{-\alpha t} - e^{-bt})\right)$	Taken as comparing model for DS-2 and DS-3.
M15	Fault removal model (2003) [46]	$m(t) = \frac{a}{p-\beta}\left\{1 - \left(\frac{(1+\alpha)e^{-bt}}{1+\alpha e^{-bt}}\right)^{\frac{c}{b}(p-\beta)}\right\}$	Taken as comparing model for DS-2 and DS-3.
M16	SRGM-3 model (2011) [47]	$m(t) = \frac{A}{1-\alpha}\left[1 - \left(\left(1 + bt + \frac{b^2t^2}{2}\right)e^{-bt}\right)^{p(1-\alpha)}\right]$	Taken as comparing model for DS-2 and DS-3.
M17	FDP: Weibull-type testing coverage FCP: with logistic r(t)	$m_d(t) = \frac{a}{1-\alpha}\left(1 - e^{-c(1-\alpha)t^r}\right)$ $m_c(t) = \frac{a}{1-\alpha}\left(1 - e^{-c(1-\alpha)t^r}\right)\frac{1}{1+be^{-\beta t}}$	Proposed model I.
M18	FDP: Delayed S-shaped testing coverage FCP: with logistic r(t)	$m_d(t) = \frac{a}{1-\alpha}\left(1 - (1 + rt)^{1-\alpha}e^{-r(1-\alpha)t}\right)$ $m_c(t) = \frac{a}{1-\alpha}\left(1 - (1 + rt)^{1-\alpha}e^{-r(1-\alpha)t}\right) \cdot \frac{1}{1+be^{-\beta t}}$	Proposed model II.
M19	FDP: Inflection S-shaped testing coverage FCP: with logistic r(t)	$m_d(t) = \frac{a}{1-\alpha}\left(1 - \left(\frac{(1+c)e^{-rt}}{1+ce^{-rt}}\right)^{1-\alpha}\right)$ $m_c(t) = \frac{a}{1-\alpha}\left(1 - \left(\frac{(1+c)e^{-rt}}{1+ce^{-rt}}\right)^{1-\alpha}\right) \cdot \frac{1}{1+be^{-\beta t}}$	Proposed model III.

3. Parameter Estimation Methods and Model Comparison Criteria

This section presents two kinds of parameter estimation methods and model comparison criteria for paired failure data and single process failure data, respectively. Paired failure data refers to observations of both fault detection and correction, and single process failure data refers to observations of only the detected fault number. This is because models for bother fault detection and correction are described as paired models. In addition to those parameters for fault detection, there are parameters for fault correction, therefore all parameters should be estimated together. For a long time in the software reliability modeling field, few published datasets were available, including both observations of fault detection and correction; it is only in recent years that considerable efforts have been made to collect more and more data including both observations of fault detection and correction in software projects, which greatly support the research of modeling and the analysis of dual processes. Conversely, parametric SRGM research depends on failure data gathered during software testing, that is, when different types of failure data are presented, one needs to adopt the corresponding method in terms of their characteristics.

3.1. Parameter Estimation Methods

3.1.1. Parameter Estimation Method for Paired FDP and FCP Models

Against failure observations of both fault detection and correction, the sum of the squared residuals for both detected and corrected faults is shown as follows:

$$LL = \left[\sum_{i=1}^{n} (y_{di} - \hat{m}_d(t_i))^2 + \sum_{i=1}^{n} (y_{ci} - \hat{m}_c(t_i))^2 \right] \tag{12}$$

where y_{di} is the cumulative number of detected faults observed until time t_i. $\hat{m}_d(t_i)$ is the estimated cumulative number of detected faults until time t_i obtained from the fitted mean value function. y_{ci} is the cumulative number of corrected faults until time t_i, and $\hat{m}_c(t_i)$ is the estimated cumulative number of corrected faults until time t_i obtained from the fitted mean value function, $i = 1, 2, \ldots, n$.

Obviously, the parameters of the model need to minimize the sum of squared deviations of detected defects and corrected defects at the same time. Here we take the combined least square estimation (LSE) method to estimate the models' parameters. The solution of simultaneous equations can be obtained by calculating the derivatives of each parameter in Equation (12) and setting the result to be equal to zero. Taking Equation (7) as an example, the following is

$$\left\{ \begin{array}{l} \frac{\partial LL}{\partial a} = \sum\limits_{i=1}^{n} \left(\frac{\partial \hat{m}(t_{di})}{\partial a}(y_{di} - \hat{m}(t_{di})) + \frac{\partial \hat{m}(t_{ci})}{\partial a}(y_{ci} - \hat{m}(t_{ci})) \right) = 0 \\ \frac{\partial LL}{\partial b} = \sum\limits_{i=1}^{n} \left(\frac{\partial \hat{m}(t_{di})}{\partial b}(y_{di} - \hat{m}(t_{di})) + \frac{\partial \hat{m}(t_{ci})}{\partial b}(y_{ci} - \hat{m}(t_{ci})) \right) = 0 \\ \frac{\partial LL}{\partial \alpha} = \sum\limits_{i=1}^{n} \left(\frac{\partial \hat{m}(t_{di})}{\partial \alpha}(y_{di} - \hat{m}(t_{di})) + \frac{\partial \hat{m}(t_{ci})}{\partial \alpha}(y_{ci} - \hat{m}(t_{ci})) \right) = 0 \\ \frac{\partial LL}{\partial d} = \sum\limits_{i=1}^{n} \left(\frac{\partial \hat{m}(t_{di})}{\partial d}(y_{di} - \hat{m}(t_{di})) + \frac{\partial \hat{m}(t_{ci})}{\partial d}(y_{ci} - \hat{m}(t_{ci})) \right) = 0 \\ \frac{\partial L}{\partial \beta} = \sum\limits_{i=1}^{n} \left(\frac{\partial \hat{m}(t_{di})}{\partial \beta}(y_{di} - \hat{m}(t_{di})) + \frac{\partial \hat{m}(t_{ci})}{\partial \beta}(y_{ci} - \hat{m}(t_{ci})) \right) = 0 \\ \frac{\partial L}{\partial r} = \sum\limits_{i=1}^{n} \left(\frac{\partial \hat{m}(t_{di})}{\partial r}(y_{di} - \hat{m}(t_{di})) + \frac{\partial \hat{m}(t_{ci})}{\partial r}(y_{ci} - \hat{m}(t_{ci})) \right) = 0 \end{array} \right. \tag{13}$$

After solving these equations numerically, we can obtain the point estimates of all parameters for the proposed model.

3.1.2. Parameter Estimation Method for Single Process Models

For observations with only detected faults, the sum of the squared distance is shown as follows to obtain the least square estimates:

$$L = \sum_{i=1}^{n} (y_i - \hat{m}(t_i))^2 \tag{14}$$

where y_i denotes the cumulative detected fault number until time t_i.

Similarly, from Equation (14), we can obtain the corresponding equations to solve estimators, taking FCP in Equation (7) for example:

$$\begin{cases} \frac{\partial L}{\partial a} = \sum\limits_{i=1}^{n} \frac{\partial \hat{m}(t_i)}{\partial a}(y_i - \hat{m}(t_i)) = 0 \\ \frac{\partial L}{\partial b} = \sum\limits_{i=1}^{n} \frac{\partial \hat{m}(t_i)}{\partial b}(y_i - \hat{m}(t_i)) = 0 \\ \frac{\partial L}{\partial \alpha} = \sum\limits_{i=1}^{n} \frac{\partial \hat{m}(t_i)}{\partial \alpha}(y_i - \hat{m}(t_i)) = 0 \\ \frac{\partial L}{\partial d} = \sum\limits_{i=1}^{n} \frac{\partial \hat{m}(t_i)}{\partial d}(y_i - \hat{m}(t_i)) = 0 \\ \frac{\partial L}{\partial \beta} = \sum\limits_{i=1}^{n} \frac{\partial \hat{m}(t_i)}{\partial \beta}(y_i - \hat{m}(t_i)) = 0 \\ \frac{\partial L}{\partial r} = \sum\limits_{i=1}^{n} \frac{\partial \hat{m}(t_i)}{\partial r}(y_i - \hat{m}(t_i)) = 0 \end{cases} \tag{15}$$

After solving equations in (15) simultaneously, we can derive the least square estimators of all model parameters.

3.2. Criteria for a Comparison of the Power of Models with Paired FDP and FCP Models

3.2.1. Criteria for a Comparison of the Descriptive Power of Models with Paired FDP and FCP Models

Under the context of the LSE method to obtain the estimates of all parameters, the combined mean squared errors for both fault detection and correction are taken as the measurement to examine the fitting performance of paired models, which is defined as [6]

$$\text{MSE} = \frac{1}{2n}\left[\sum_{i=1}^{n}(y_{di} - \hat{m}_d(t_i))^2 + \sum_{i=1}^{n}(y_{ci} - \hat{m}_c(t_i))^2\right] \tag{16}$$

Thus, the lower value of MSE indicates the better goodness-of-fit performance.

3.2.2. Criteria for a Comparison of the Predictive Power of Models with Paired FDP and FCP Models

Mean relative errors (MREs) for both fault detection and correction processes are taken as a criterion to examine the prediction performance of paired models. MRE is expressed as follows:

$$\text{MRE} = \frac{1}{2(n-j+1)}\left(\sum_{i=j}^{n}\left|\frac{\hat{m}_d(t_i) - y_{di}}{y_{di}}\right| + \sum_{i=1}^{n}\left|\frac{\hat{m}_c(t_i) - y_{ci}}{y_{ci}}\right|\right) \tag{17}$$

Assume that by the end of testing time t_n, totally paired data (y_{dn}, y_{cn}) are collected. Firstly, we use the paired data up to time $t_{j-1}(t_{j-1} < t_j < t_n)$ to estimate the parameters of $\hat{m}_d(t)$ and $\hat{m}_c(t)$, then substitute the estimated parameters into the mean value functions to yield the prediction values of the cumulative fault numbers $\hat{m}_d(t_j)$ and $\hat{m}_c(t_j)$. Then the procedure is repeated from t_j to t_n.

Therefore, the fewer MREs, the better the model's prediction performance.

3.3. Criteria for a Comparison of the Power of Model with Single Process Models

3.3.1. Criteria for a Comparison of the Descriptive Power of Models with Single Process Models

For single process models, we adopted the following seven goodness-of-fit criteria to compare their fitting performance.

(1) Mean value of squared error (MSE)

$$\text{MSE} = \frac{1}{n-N}\sum_{i=1}^{n}(y_i - \hat{m}(t_i))^2 \tag{18}$$

where N represents the number of parameters in the model. Thus, the lower MSE indicates a better goodness-of-fit.

(2) Correlation index of the regression curve equation (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{m}(t_i))^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{19}$$

where $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Therefore, the larger R^2 means the better model.

(3) Adjusted R^2

$$\text{Adjusted } R^2 = 1 - \frac{(1-R^2)(n-1)}{n-P-1} \tag{20}$$

where P represents the number of predictors in the model. The larger the Adjusted R^2, the smaller the fitting error.

(4) Predictive-ratio risk (PRR)

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

The lower the value of PRR, the better the goodness-of-fit.

(5) Predictive power (PP)

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

Less PP means a better fitting.

(6) *Bias*

$$Bias = \frac{1}{n}\sum_{i=1}^{n}(\hat{m}(t_i) - y_i) \tag{23}$$

The lower *Bias* indicates the preferred model.

(7) *Variation*

$$Variation = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \hat{m}(t_i) - Bias)^2} \tag{24}$$

The *Variation* with a lower value has a better goodness-of-fit.

3.3.2. Criteria for a Comparison of the Predictive Power of Models with Single Process Models

For single process models, we used the SSE criterion to examine the predictive power of SRGMs. SSE is expressed as follows:

$$\text{SSE} = \sum_{i=m}^{n}(y_i - \hat{m}(t_i))^2 \tag{25}$$

A lower SSE means a better prediction performance.

4. Numerical Examples

In the following experiments, we validate the proposed models on three real datasets and compare their performance based on different comparison criteria.

4.1. Case Study 1

DS-1 is collected from testing a medium size software system [10] and has been widely used in many papers, such as [7,11,23,48]. DS-1 includes observations of not only detected faults but also corrected faults, in which there was a total of 144 faults observed and 143 faults corrected within 17 weeks. For simplicity and tractability, the detailed information about this dataset can be referenced in [6]. We chose M1 to M5 (paired FDP and FCP models) as comparing models, meanwhile the combined LSE method and combined MSE and MRE values were chosen as the parameter estimation method and goodness-of-fit and prediction performance criteria, respectively.

For the descriptive power comparison, all data points were used to fit the models and estimate the model parameters. The results are listed in Table 3, together with the MSE values used for a goodness-of-fit comparison.

Table 3. Estimates of parameters and performance values of paired models for DS-1.

Models	Parameter Estimation Values						Estimation	Prediction
	a	b	c/μ	α/σ	β	r	MSE	MRE
M1	157.6607	0.1353	1.5813	-	-	-	MSE = 52.6776 MSEd = 49.5562 MSEc = 55.7991	MRE = 0.2667 MREd = 0.3067 MREc = 0.2267
M2	168.3627	0.1193	0.0279	-	-	-	MSE = 104.8889 MSEd = 58.0583 MSEc = 151.7194	MRE = 0.3422 MREd = 0.4743 MREc = 0.2102
M3	156.3453	0.1404	0.5811	-	-	-	MSE = 55.1920 MSEd = 50.6615 MSEc = 59.7225	MRE = 0.2611 MREd = 0.3083 MREc = 0.2140
M4	152.6053	0.1501	1.9756	0.3050	-	-	MSE = 40.8773 MSEd = 52.6564 MSEc = 29.0983	MRE = 0.2030 MREd = 0.2422 MREc = 0.1639
M5	152.2418	0.1466	-	1.7071	0.6081	-	MSE = 34.6214 MSEd = 49.4108 MSEc = 19.8320	MRE = 0.2018 MREd = 0.2171 MREc = 0.1865
M17	99.9970	27.8834	0.1601	0.2805	0.8266	1.2048	MSE = **27.9714** MSEd = 37.6473 MSEc = 18.2954	MRE = **0.0980** MREd = 0.1043 MREc = 0.0917
M18	99.9888	7.2904	-	0.2538	0.5268	0.4833	MSE = **27.5294** MSEd = 29.3943 MSEc = 25.6646	MRE = **0.0475** MREd = 0.0495 MREc = 0.0455
M19	15.7243	15.2734	5.9790	0.8871	0.6848	1.8583	MSE = **22.9948** MSEd = 26.6022 MSEc = 19.3873	MRE = **0.0921** MREd = 0.0981 MREc = 0.0861

Notes: The bold numbers mean the top 3 best results of models in this column.

For the predictive performance comparison, we used the first 80% of DS-1 to estimate the parameters of all models, then we compared the prediction ability of all models according to the remaining 20% of the data points. The results of the MRE values are listed in the far right column of Table 3.

From Table 3, it can be noted that for the proposed models:

- M19 (the proposed model with inflection S-shaped testing coverage) has the smallest MSE = 22.9948 and MSEd = 26.6022 among all models.

- M18 (the proposed model with delayed S-shaped testing coverage) has the second smallest MSE = 27.5294 and MSEd = 29.3943 among all models.
- M17 (the proposed model with Weibull-type testing coverage) has the third smallest MSE = 27.9714 and MSEd = 37.6473 among all models.

According to the combined MSE criterion (16), all proposed models, that is, M17, M18 and M19, have a better goodness-of-fit performance than existing models. We also noticed that the proposed models' MSE values belong to value interval of (22.9948, 27.9714), whereas existing models' MSE values belong to interval (34.6214, 104.8889), apparently the proposed models' MSE values are much smaller than the values of existing models, e.g., other models' MSE values can be 1.53 times (G-O with gamma distributed time delay model (M5)'s 34.6214) and even 4.56 times (G-O with time-dependent delay model (M2)'s 104.8889) larger than the value of the proposed model of M19. So, we can deduce that for DS-1, the descriptive power of our proposed models based on the fault amount dependency between fault detection and fault correction processes is better than those of other models based on the time dependency between these two processes.

Figure 2a–h shows the graph of the fitting comparison of existing paired models M1 to M5 with the proposed models M17 to M19 based on DS-1. From Figure 2, we can see that the proposed models fit the dataset very well in both fault detection and correction processes. Especially, Figure 2b–d shows that the fitting curves of the existing models M2, M3, and M1 have a great deviation from the curves of the actual fault data, whose fitting degree are far less than those of the models proposed in this paper.

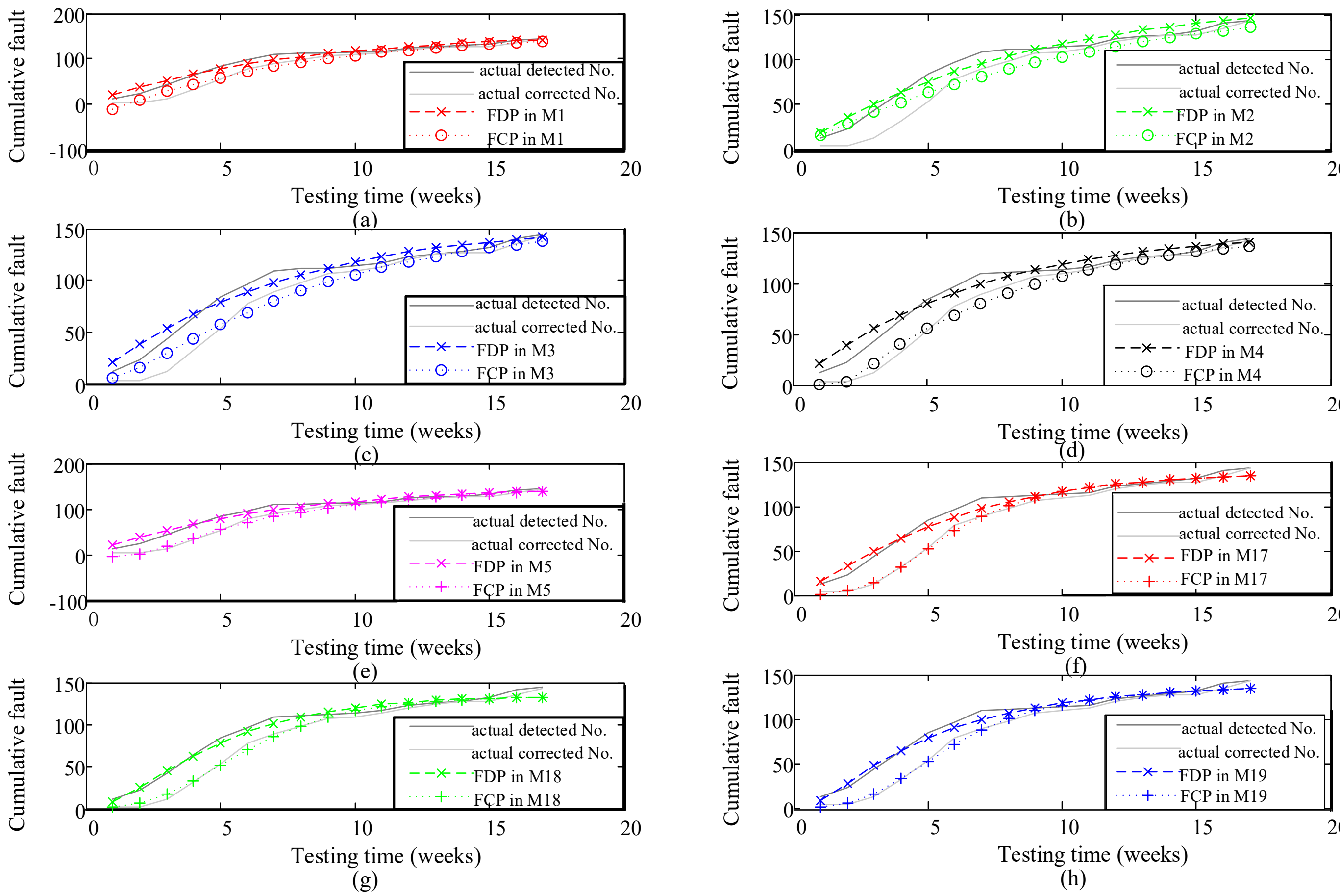

Figure 2. Comparisons of the fitting results based on M1-M5 and M17-M19 for DS-1 (**a**–**e**) FDP and FCP fitting results based on M1-M5 vs. actual detected faults and corrected fault number; (**f**–**h**) FDP and FCP fitting results based on M17-M19 vs. actual detected faults and corrected fault number.

For the predictive power comparison, similarly, all proposed models, that is, M17, M18, and M19, provide better predictive power than the existing models. M18 presents the smallest MRE = 0.0475 followed by M19's MRE = 0.0921, which is the second smallest value of MRE and M17's MRE = 0.0980, which is the third smallest value of MRE. Other existing models' MRE values belong to interval (0.2018, 0.3422), which is obviously much larger than the value range of the proposed models, e.g., other MRE values can be 4.25 times (G-O with gamma distributed time delay model (M5)'s 0.2018) and even 7.2 times (G-O with time-dependent delay model (M2)'s 0.3422) larger than the value of the proposed model M18.

Figure 3 shows the graph of the prediction comparison of all models based on DS-1, which agrees with those in Table 3. Obviously, it shows that the proposed models have the best predictive performance through the intuitive effect of the eyes. Overall, MRE values of M1 to M5 are far greater than those of M17 to M19. Numerically, MRE values of M1 to M5 are almost two to three times that of M17 to M19. Among them, the MRE value of M2 is the largest, which is close to 0.5, while that of M18 is the smallest, namely 0.05. In other words, the prediction accuracy of M2 is almost one order of magnitude lower than that of M18.

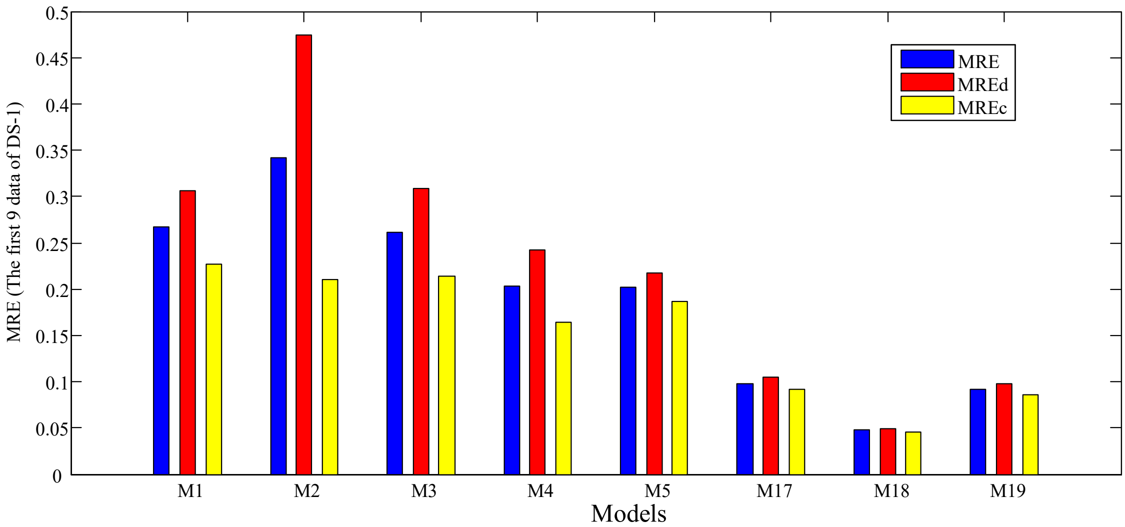

Figure 3. Comparisons of the prediction results based on M1-M5 and M17-M19 for DS-1.

4.2. Case Study 2

Dataset 2 (DS-2) is collected from Tandem Computer Release #1 [49] and has been used in papers, such as [46,50]. There was a total of 20 faults observed within about 20 weeks, which includes observations of only fault detection data. Therefore, we chose M6 to M16 (single process models) as comparing models, meanwhile the traditional LSE method (14) and seven criteria (18)–(24) were chosen as the parameter estimation method and goodness-of-fit performance criteria.

For the descriptive power comparison, all data points were used to fit the models and estimate model parameters. The estimates of model parameters and all seven estimation criteria values are listed in Tables 4 and 5, respectively.

Table 4. Parameter estimates of M6 to M19 for DS-2.

Models	Parameter Estimation Values					
	$a(A)$	b	c	α	β	$r(p)$
M6	130.2	8.317×10^{-2}	-	-	-	-
M7	104.0	0.2654	-	-	-	-
M8	110.8	0.1721	1.205	-	-	
M9	999.5	-	-	0.51	7.685×10^{-2}	0.279
M10	115.8	-	-	0.6548	1.721×10^{-2}	3.03
M11	121.1	-	-	245.2	3.542×10^{-4}	1.027
M12	130.2	8.317×10^{-2}	-	9.363×10^{-10}	-	-
M13	130.2	8.317×10^{-2}	-	1.04×10^{-9}	-	-
M14	1.589×10^{-8}	0.1721	110.8	3.68×10^{-4}	1.205	-
M15	103.6	8.193×10^{-2}	4.916	57.49	4.827×10^{-5}	0.9993
M16	83.46	37.1	-	0.3433	-	3.678×10^{-3}
M17	155.1	5.048	0.6796	6.488×10^{-4}	0.2798	0.1665
M18	98.05	6.62	-	0.05596	0.2849	32.76
M19	104.1	6.468	167.7	2.166×10^{-6}	0.2813	7.396

Table 5. Comparisons of descriptive and predictive power of M6 to M19 for DS-2.

Models	Descriptive Power Criteria Values							Predictive Power
	MSE	R^2	Adjusted R^2	PRR	PP	Bias	Variation	SSE (85% of DS-2)
M6	12.9056	0.9857	0.9849	0.3783	0.2028	−0.0895	3.5005	223.8406
M7	28.0611	0.9689	0.9672	19.5655	1.0809	−1.4056	5.7293	**1.8720**
M8	10.5647	0.9890	0.9877	0.8682	0.3049	−0.4480	3.1758	159.9882
M9	14.8438	0.9854	0.9827	0.3659	0.2002	−0.0972	3.5400	72.5264
M10	49.4188	0.9514	0.9422	57.1993	1.4971	−2.0420	7.4017	46.3688
M11	15.1750	0.9851	0.9823	0.8342	0.3007	−0.3531	3.4487	270.4970
M12	13.6647	0.9857	0.9849	0.3783	0.2028	−0.0895	3.5005	223.8409
M13	13.6647	0.9857	0.9849	0.3783	0.2028	−0.0895	3.5005	223.8407
M14	11.9733	0.9890	0.9860	0.8682	0.3049	−0.4480	3.1758	111.9241
M15	10.8786	0.9906	0.9873	0.8766	0.3036	−0.4914	2.9627	108.1806
M16	14.9250	0.9853	0.9826	0.7103	0.2787	−0.2323	3.5698	5.7293×10^3
M17	**3.3221**	**0.9971**	**0.9961**	**0.0242**	**0.0215**	**−0.0701**	**1.5696**	**19.3357**
M18	**2.1653**	**0.9980**	**0.9975**	**0.0224**	**0.0224**	**−0.0164**	**1.3077**	19.6887
M19	**2.0100**	**0.9981**	**0.9975**	**0.0135**	**0.0131**	**−0.0268**	**1.2607**	**19.6620**

Notes: The bold numbers mean the top 3 best results of SRGMs in this column.

For the predictive performance comparison, we used the first 85% of DS-2 to estimate the parameters of all models, and then we compared the prediction ability of all models according to the remaining 15% of DS-2 data points. SSE criterion (25) is taken as the predictive power criterion, and the prediction values are listed in the far right column of Table 5.

Figure 4 graphically illustrates the fitting comparisons of all single process models M6 to M16 and FCPs of the proposed models M17 to M19 based on DS-2. In order to obtain a clearer display effect, we divided these models into two groups, and each group included the actual data and the proposed models. Figure 4 shows that the curves of the proposed models M17 to M19 have less deviation from the curve of the actual data than existing models, which confirms that the fitting performance of the proposed models are better than existing models.

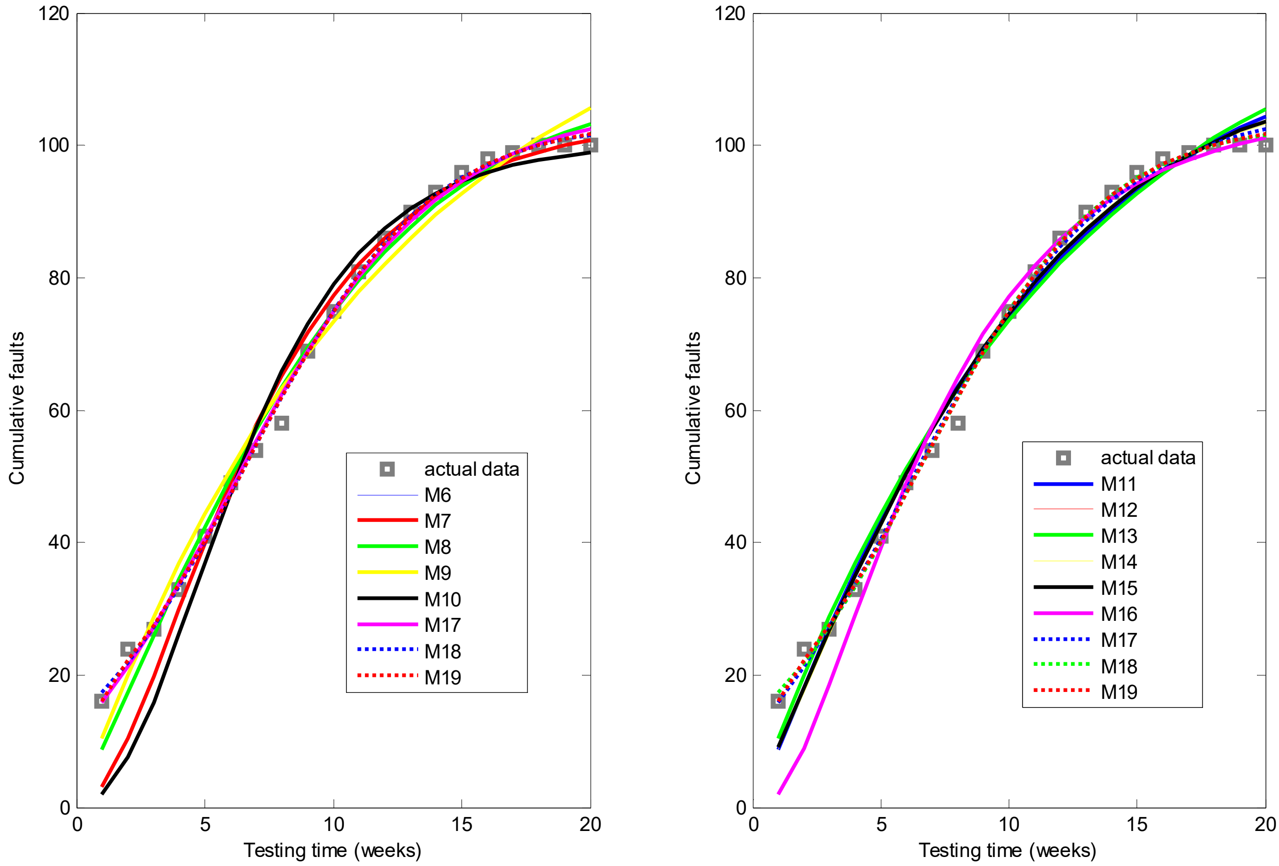

Figure 4. Comparisons of the fitting results for M11-M19 based on DS-2.

From Table 5, it is clear that all FCPs of the proposed models provide the top three best results of SRGMs according to these seven criteria values of MSE, R^2, Adjusted R^2, PRR, PP, $Bias$, and $Variation$.

- M17 (the proposed model with Weibull-type testing coverage), M18 (the proposed model with delayed S-shaped testing coverage), and M19 (the proposed model with inflection S-shaped testing coverage) provide much smaller MSE values than existing models, among which M19's MSE = 2.0100 is the lowest value among all models followed by M18's MSE = 2.1653 and M17's MSE = 3.3221.
- M17, M18, and M19 provide the largest R^2 values of 0.9971, 0.9980, and 0.9981, respectively, compared to existing models, where M19 provides the largest R^2 = 0.9981.
- M17, M18, and M19 provide the largest Adjusted R^2 values of 0.9960, 0.9975, and 0.9975, respectively, compared to existing models, where M18 and M19 provide the largest R^2 = 0.9975.
- M17, M18, and M19 provide the smallest PRR values of 0.0242, 0.0224, and 0.0135, respectively, compared to existing models, where M19 provides the smallest PRR = 0.0135 followed by M18's 0.0224 and M17's 0.0242.
- M17, M18, and M19 provide the smallest PP values of 0.0215, 0.0224, and 0.013, respectively, compared to existing models, where M19 provides the smallest PP = 0.0131 followed by M17's 0.0215 and M18's 0.0224.
- M17, M18, and M19 provide the smallest absolute values of $Bias$ of 0.0701, 0.0164, and 0.02681, respectively, compared to existing models, where M18 provides the smallest absolute value of $Bias$ = 0.0164 followed by M19 and M17.
- M17, M18, and M19 provide the smallest $Variation$ values of 1.5696, 1.3077, and 1.2607, respectively, compared to existing models, where M19 provides the smallest $Variation$ = 1.2607 followed by M18 and M17.

To sum up, among these three proposed models, M19 affords the best results for MSE, R^2, Adjusted R^2, PRR, PP, and $Variation$; and M18 provides the best result for $Bias$. Therefore, according to the comparison results in this dataset, we can conclude that the proposed models seem to have a better fitting performance overall.

For the predictive power comparison, though the SSE values given by the proposed models are not the smallest results, they are the second, third, and fourth best results, respectively, which are slightly bigger than the best value. That is, since the minimum SSE value is 1.8720 (given by the delayed S-shaped model (M7)), and compared to other models, we note that the SSE values given by the proposed models are far better than those of other models. For example, the SSE values of other models can be 2.4 times (given by the Yamada Rayleigh model (M10), whose value equals to 46.3688) or even 296.31 times (given by the SRGM-3 model (M16), whose value equals to 5.7293×10^3) that of the proposed model M17, namely 19.3357. In addition, the delayed S-shaped model only affords the best result for DS-2 but does not afford the best result for DS-3, meanwhile it does not give the best goodness-of-fit result for DS-2. Among these three proposed models, M17 gives the best prediction result followed by M19 and M18. Their SSE values belong to interval (19.3357, 19.6887), which indicates that for DS-2, the proposed models show the same level of prediction accuracy.

Figure 5 shows the graph of the cumulative number of detected faults for all models using 85% of DS-2, which agrees with those in Table 5. Obviously, it shows that the proposed models have a better predictive performance except M7, but far better than other models; e.g., M16 has a large deviation from the curve of the actual data than all models.

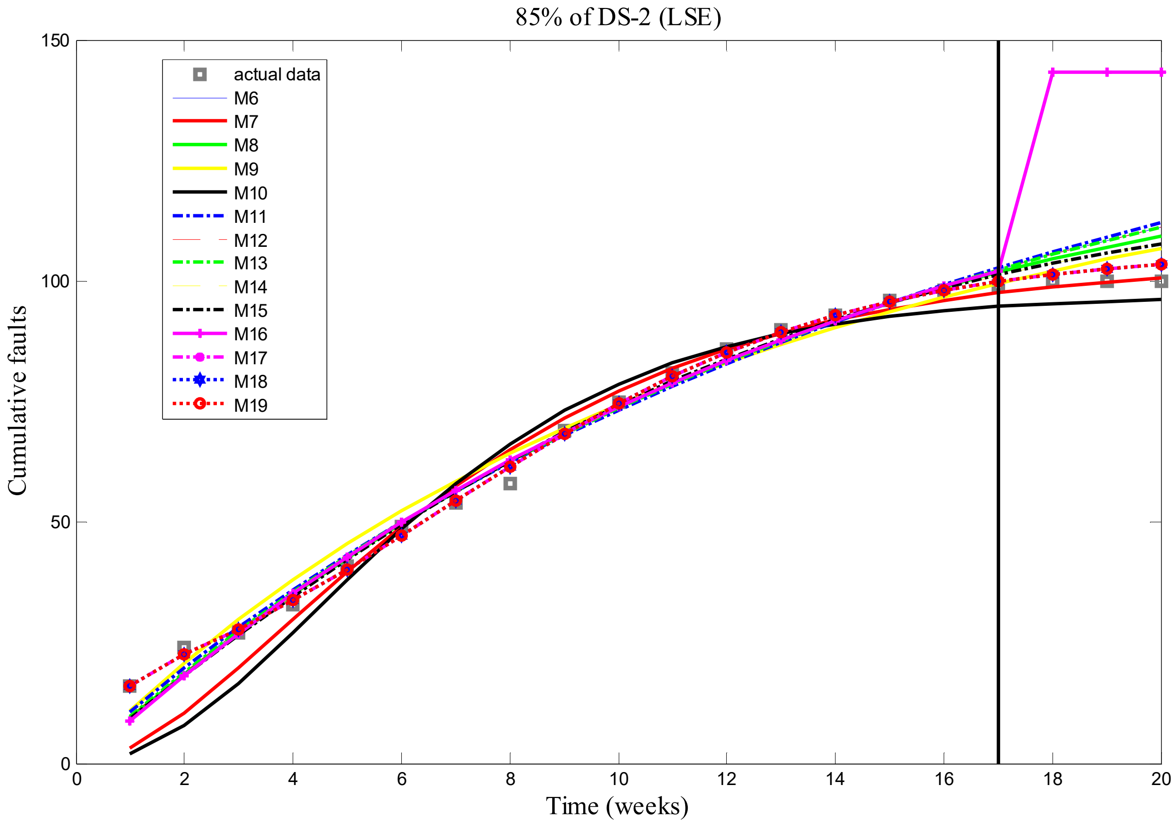

Figure 5. Comparisons of the prediction results for M11-M19 based on DS-2.

4.3. Case Study 3

Dataset 3 (DS-3) is from a networking component of the Linux Kernel project [51] and has been widely used in many papers, such as [52]. Linux is a common open-source operating system, which is popular and represents the actual situation of the current software system to a certain extent, and the failure dataset is collected from Bugzilla. There was a total of 2251 faults observed within 193 months (5933 days) from 15 November 2002 to 28 January 2019. This dataset also has observations of only fault detection data, and compared to DS-1 and DS-2 this dataset has a larger time series.

For the descriptive power comparison, we utilized all data points to fit the models and estimate model parameters. Similarly, the estimated values of model parameters are tabulated in Table 6. Seven criteria values for fitting performance comparisons are summarized in Table 7, along with the prediction comparison results of SSE in the far right column. The fitting of the models on DS-3 is graphically illustrated in Figure 6. As is clearly shown, the curves of the proposed models M17 to M19 are much closer to the curve of the actual data than existing models, which confirms that the fitting performance of the proposed models is much better than existing models.

Table 6. Parameter estimates of M6 to M19 for DS-3.

Models	**Parameter Estimation Values**					
	***a*(*A*)**	***b***	***c***	***α***	***β***	***r*(*p*)**
M6	1.177×10^6	9.361×10^{-6}	-	-	-	-
M7	3.863×10^3	1.004×10^{-2}	-	-	-	-
M8	3.142×10^3	1.619×10^{-2}	7.644	-	-	-
M9	5.479×10^5	-	-	0.9798	2.129×10^{-5}	0.9662
M10	4.684×10^3	-	-	0.9532	6.679×10^{-5}	0.9193
M11	7.692×10^3	-	-	0.8348	1.084×10^{-4}	1.519
M12	8.748×10^3	9.186×10^{-4}	-	4.776×10^{-3}	-	-
M13	2.2×10^3	3.188×10^{-3}	-	1.239×10^{-2}	-	-
M14	0.2537	1.618×10^{-2}	3.143×10^3	35.02	7.637	-
M15	7.031×10^3	3.903×10^{-2}	1.857×10^{-3}	0.9799	0.3611	0.6906
M16	6.262×10^3	0.1255	-	0.7208	-	2.079×10^{-2}
M17	5.652×10^3	32.29	0.00133	0.6152	9.54×10^{-3}	1.874
M18	194.7	1.729	-	1.005	0.05865	0.07327
M19	6.571×10^3	4.546	3.642	0.3119	0.05695	6.152×10^{-3}

Table 7. Comparisons of descriptive and predictive power of M6 to M19 for DS-3.

Models	**Descriptive Power Criteria Values**							**Predictive Power**
	MSE	R^2	**Adjusted** R^2	**PRR**	**PP**	**Bias**	**Variation**	**SSE (90% of DS-3)**
M6	1.3628×10^4	0.9728	0.9726	20.4143	1.6392×10^3	50.5289	146.0383	6.4339×10^5
M7	1500.0000	0.9970	0.9970	**3.2748**	1.9292	−5.5046	39.8940	1.1006×10^5
M8	1.7321×10^3	0.9966	0.9965	9.7466	428.2829	6.8714	43.1907	7.6730×10^4
M9	1.3926×10^4	0.9725	0.9720	20.4693	1.6471×10^3	50.7397	146.7860	1.1939×10^6
M10	3.1280×10^3	0.9938	0.9937	13.3523	4.7410	−12.3131	59.6022	3.9461×10^5
M11	1.0963×10^3	0.9978	0.9978	4.1251	32.5139	**0.4720**	33.0260	6.2463×10^4
M12	3.5326×10^3	0.9930	0.9929	13.7110	831.6408	12.6471	63.2145	2.8999×10^5
M13	2.5505×10^3	0.9949	0.9949	12.1011	628.9523	11.0480	53.8978	1.5606×10^5
M14	1.7505×10^3	0.9965	0.9964	9.7538	428.8446	6.9531	43.2306	7.4894×10^4
M15	4.6925×10^3	0.9908	0.9906	13.0302	576.6723	18.5959	75.0555	$\mathbf{4.2218 \times 10^4}$
M16	952.3810	0.9981	0.9981	10.1299	**1.8643**	**0.1743**	30.6975	$\mathbf{5.3360 \times 10^3}$
M17	**827.8075**	**0.9984**	**0.9983**	3.9069	**1.7518**	−0.7217	**3.2352**	6.3205×10^4
M18	**901.0638**	**0.9982**	**0.9982**	**2.8199**	**1.5354**	−1.6940	**29.5908**	$\mathbf{6.0526 \times 10^3}$
M19	**856.6845**	**0.9983**	**0.9983**	**3.1131**	22.5339	**−0.4106**	**28.9657**	4.8129×10^4

Notes: The bold numbers mean the top 3 best results of SRGMs in this column.

Table 7 shows that:

- The proposed models provide the top three best results over all models according to the estimation criteria values of MSE, R^2, Adjusted R^2, and *Variation*, where M17 (the proposed model with Weibull-type testing coverage) gives the best results of MSE, R^2, Adjusted R^2, and *Variation* followed by M19 (the proposed model with inflection S-shaped testing coverage).

- M18 (the proposed model with delayed S-shaped testing coverage) gives the best results of PRR and PP.
- Though M18 gives the second-best result of the absolute value of $Bias$, whereas the SRGM-3 model gives the best result, there is a small difference between these two results, that is, the lowest absolute value of $Bias$ is 0.1743, and the absolute value of $Bias$ given by M18 is 0.4106.

Therefore, according to the above results, the proposed models generally provide a better fitting performance.

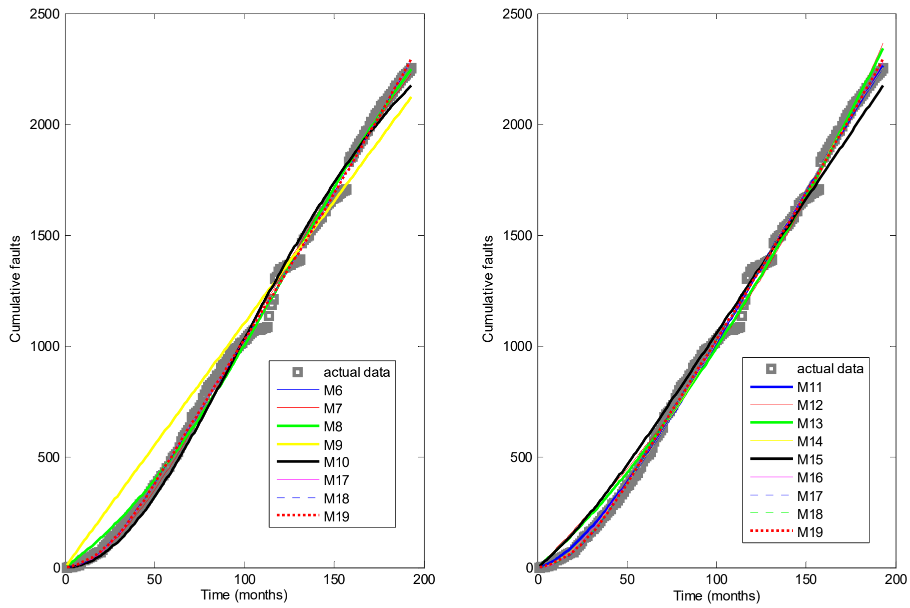

Figure 6. Comparisons of the fitting results for M1-M5 and M17-M19 based on DS-3.

For the predictive performance comparison, we used the first 90% of DS-3 to estimate the model parameters, and then we adopted the remaining 10% of the DS-3 data points to compare the prediction ability of the models. Though the values provided by the proposed models are not the best results, M18 presents the second-best result, which is slightly bigger than the best result, for the smallest SSE value is 5.3360×10^3 (given by SRGM-3 model), and M18's SSE value is 6.0526×10^3. Since DS-3 has a larger time series, the prediction accuracy of M18 is in the same order of magnitude as that of the SRGM-3 model, which means that the prediction accuracy of the two models is almost the same. Compared to other models, we find that the proposed models' SSE values belong to interval (6.0526×10^3, 6.3205×10^4), whereas the existing models' SSE values belong to interval (5.3360×10^3, 1.1939×10^6), and the proposed models' SSE values are relatively smaller than the values of existing models; e.g., the other models' SSE values can be 6.98 times (Fault removal model's 4.2218×10^4) and even 197.25 times (Yamada exponential model's 1.1939×10^6) larger than the value of the proposed model of M18. In addition, for the SRGM-3 model, it only provides the best prediction result for DS-3, but provides the worst prediction result for DS-2. Overall, the proposed models give a better predictive performance.

Figure 7 shows the graph of the cumulative number of detected faults for all models using 90% of DS-3, which agrees with those in Table 7. Obviously, it shows that the proposed models have a better predictive performance in general.

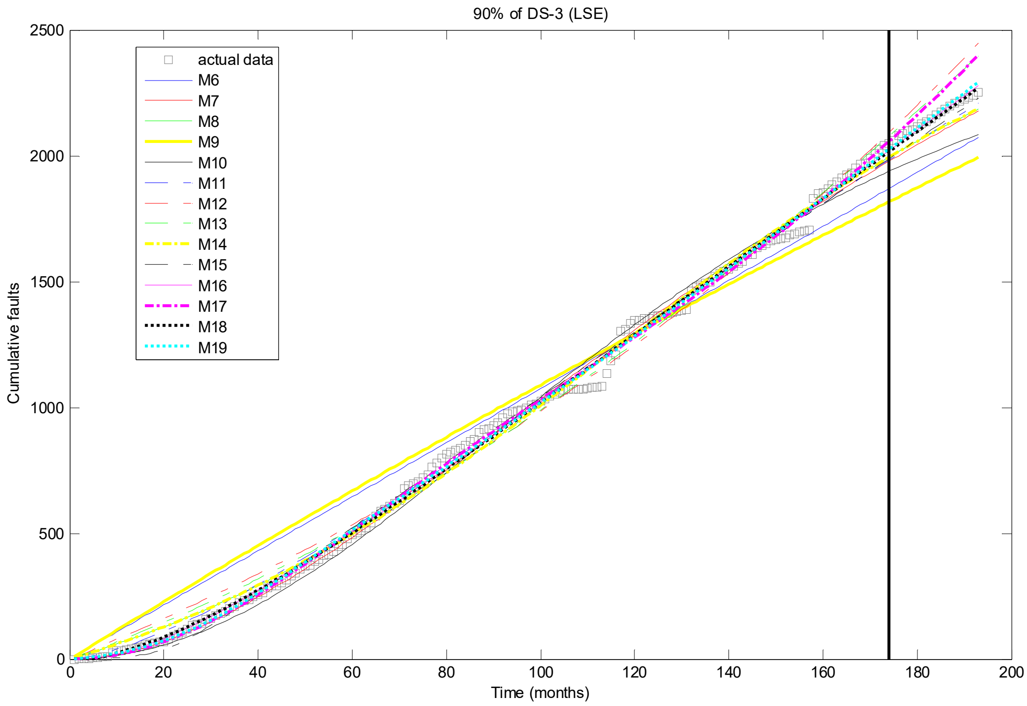

Figure 7. Comparisons of the prediction results of M6 to M19 for DS-3.

5. Conclusions

In this article, we integrated three testing coverage functions (Weibull-type, delayed S-shaped, and inflection S-shaped) into software reliability modeling based on NHPP with incorporation of both fault detection processes and fault correction processes. The relationship between the mean value functions of detected faults and corrected faults is introduced from the viewpoint of fault amount dependency instead of time dependency. We compared the performance of three proposed models with several existing models on three real failure datasets in which two kinds of failure data are employed. For one kind of failure data that contained only the detected fault number, the traditional LSE method was used to estimate the model parameters, and seven comparison criteria were adopted to compare the fitting performance together with the SSE value to compare the prediction performance of models. For another kind of failure data that contained failure observations of both fault detection and correction, the combined LSE method was used for parameter estimation, and the combined MSE and MRE values were used for model fitting and prediction performance comparison. No matter the case, the proposed models provided generally better goodness-of-fit and prediction results compared with other existing paired models and single process models.

In addition, the results show that models with testing coverage can generally give a better fit and prediction to the observed data, so we suggest that an extension study on time variable testing coverage functions should be performed. Moreover, we will use the maximum likelihood estimation method to estimate the parameters and their confidence intervals of the proposed models to compare any difference between limitation on data form or accuracy of model performance in the future. We will also discuss the release time problem by use of the proposed models giving reliability function simulation under various sample sizes.

Currently, we are considering more complicated circumstance, such as incorporating more different functions to characterize the relationship between the fault amounts of the two processes. We should have more failure datasets to verify such a software reliability model and to support the conclusions we made. Further progress with respect to these subjects will be proposed in the future paper.

Author Contributions: Conceptualization, Q.L. and H.P.; Data Curation, Q.L.; Formal Analysis, Q.L.; Funding Acquisition, Q.L.; Investigation, Q.L.; Methodology, Q.L. and H.P.; Project Administration, Q.L.; Resources, Q.L. and H.P.; Supervision, H.P.; Writing—Review and Editing, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly funded by "National Key Laboratory of Science and Technology on Reliability and Environmental Engineering of China", grant number "WDZC2019601A303".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the editor and referees for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Erto, P.; Giorgio, M.; Lepore, A. The Generalized Inflection S-Shaped Software Reliability Growth Model. *IEEE Trans. Reliab.* **2020**, *69*, 228–244. [CrossRef]
2. Schneidewind, N.F. Analysis of error processes in computer software. *ACM Sigplan Not.* **1975**, *10*, 337–346. [CrossRef]
3. Xie, M.; Zhao, M. The Schneidewind software reliability model revisited. In Proceedings of the Third International Symposium on Software Reliability Engineering, Research Triangle Park, NC, USA, 7–10 October 1992; pp. 184–192.
4. Schneidewind, N.F. Modelling the fault correction process. In Proceedings of the 12th International Symposium on Software Reliability Engineering, Hong Kong, China, 27–30 November 2001; IEEE Computer Society Press: Los Alamitos, CA, USA, 2001; pp. 185–190.
5. Huang, C.Y.; Lin, C.T. Software Reliability Analysis by Considering Fault Dependency and Debugging Time Lag. *IEEE Trans. Reliab.* **2006**, *55*, 436–450. [CrossRef]
6. Xie, M.; Hu, Q.P.; Wu, Y.P.; Ng, S.H. A study of the modeling and analysis of software fault-detection and fault-correction processes. *Qual. Reliab. Eng. Int.* **2007**, *23*, 459–470. [CrossRef]
7. Wu, Y.P.; Hu, Q.P.; Xie, M.; Ng, S.H. Modeling and analysis of software fault detection and correction process by considering time dependency. *IEEE Trans. Reliab.* **2007**, *56*, 629–642. [CrossRef]
8. Okamura, H.; Dohi, T. A Generalized Bivariate Modeling Framework of Fault Detection and Correction Processes. In Proceedings of the 2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE), Toulouse, France, 23–26 October 2017.
9. Wang, L.; Hu, Q.; Jian, L. Software reliability growth modeling and analysis with dual fault detection and correction processes. *Iie Trans.* **2016**, *48*, 359–370. [CrossRef]
10. Peng, R.; Li, Y.F.; Zhang, W.J.; Hu, Q.P. Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction. *Reliab. Eng. Syst. Saf.* **2014**, *126*, 37–43. [CrossRef]
11. Peng, R.; Zhai, Q. Modeling of software fault detection and correction processes with fault dependency. *Eksploat. I Niezawodn.-Maint. Reliab.* **2017**, *19*, 467–475. [CrossRef]
12. Lo, J.H.; Huang, C.Y. An integration of fault detection and correction processes in software reliability analysis. *J. Syst. Softw.* **2006**, *79*, 1312–1323. [CrossRef]
13. Yang, J.; Liu, Y.; Xie, M.; Zhao, M. Modeling and analysis of reliability of multi-release open source software incorporating both fault detection and correction processes. *J. Syst. Softw.* **2016**, *115*, 102–110. [CrossRef]
14. Yang, J.; Zhao, M.; Chen, J. ELS algorithm for estimating open source software reliability with masked data considering both fault detection and correction processes. *Commun. Stat. Theory Methods* **2021**, 1–26. [CrossRef]
15. Pachauri, B.; Kumar, A.; Dhar, J. Reliability analysis of open source software systems considering the effect of previously released version. *Int. J. Comput. Appl.* **2019**, *41*, 31–38. [CrossRef]
16. Tiwari, A.; Sharma, A. Imperfect Debugging-Based Modeling of Fault Detection and Correction Using Statistical Methods. *J. Circuits Syst. Comput.* **2021**, *30*, 2150185. [CrossRef]
17. Choudhary, C.; Kapur, P.K.; Khatri, S.K.; Muthukumar, R.; Shrivastava, A.K. Effort based release time of software for detection and correction processes using MAUT. *Int. J. Syst. Assur. Eng. Manag.* **2020**, *11*, S367–S378. [CrossRef]
18. Saraf, I.; Iqbal, J. Generalized software fault detection and correction modeling framework through imperfect debugging error generation and change point. *Int. J. Inf. Technol.* **2019**, *11*, 751. [CrossRef]
19. Kumar, V.; Khatri, S.K.; Dua, H.; Sharma, M.; Mathur, P. An Assessment of Testing Cost with Effort-Dependent FDP and FCP under Learning Effect: A Genetic Algorithm Approach. *Int. J. Reliab. Qual. Saf. Eng.* **2014**, *21*, 1450027. [CrossRef]
20. Shu, Y.; Wu, Z.; Liu, H.; Yang, X. Software reliability modeling of fault detection and correction processes. In Proceedings of the 2009 Annual Reliability and Maintainability Symposium, Fort Worth, TX, USA, 26–29 January 2009; pp. 521–526.
21. Li, Q.Y.; Pham, H. Modeling software fault-detection and fault-correction processes by considering the dependencies between fault amounts. *Appl. Sci.* **2021**, *11*, 6998. [CrossRef]

22. Li, Q.; Pham, H. NHPP software reliability model considering the uncertainty of operating environments with imperfect debugging and testing coverage. *Appl. Math. Model.* **2017**, *51*, 68–85. [CrossRef]
23. Huang, C.Y.; Kuo, S.Y.; Lyu, M.R. An assessment of testing-effort dependent software reliability growth models. *IEEE Trans. Reliab.* **2007**, *56*, 198–211. [CrossRef]
24. Cai, X.; Lyu, M.R. Software reliability modeling with test coverage experimentation and measurement with a fault-tolerant software project. In Proceedings of the 18th IEEE International Symposium on Software Reliability, Trollhattan, Sweden, 5–9 November 2007; pp. 17–26.
25. Malaiya, Y.K.; Li, M.N.; Bieman, J.M.; Karcich, R. Software reliability growth with test coverage. *IEEE Trans. Reliab.* **2002**, *51*, 420–426. [CrossRef]
26. Pham, H.; Zhang, X. NHPP software reliability and cost models with testing coverage. *Eur. J. Oper. Res.* **2003**, *145*, 443–454. [CrossRef]
27. Vouk, M.A. Using Reliability Models during Testing with Non-Operational Profiles. Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.8863&rep=rep1&type=pdf (accessed on 24 November 2021).
28. Gokhale, S.; Trivedi, K.S. A time/structure based software reliability model. *Ann. Softw. Eng.* **1999**, *8*, 85–121. [CrossRef]
29. Park, J.Y.; Lee, G.; Park, J.H. A class of coverage growth functions and its practical application. *J. Korean Stat. Soc.* **2008**, *37*, 241–247. [CrossRef]
30. Chatterjee, S.; Shukla, A. A unified approach of testing coverage-based software reliability growth modelling with fault detection probability, imperfect debugging, and change point. *J. Softw. Evol. Process* **2019**, *31*, e2150. [CrossRef]
31. Chatterjee, S.; Saha, D.; Sharma, A.; Verma, Y. Reliability and optimal release time analysis for multi up-gradation software with imperfect debugging and varied testing coverage under the effect of random field Environments. *Ann. Oper. Res.* **2021**, 1–21. [CrossRef]
32. Jia, L.; Yang, B.; Guo, S.; Park, D.H. Software reliability modeling considering fault correction process. *IEICE Trans. Inf. Syst.* **2010**, *93*, 185–188. [CrossRef]
33. Liu, Y.; Li, D.; Wang, L.; Hu, Q. A general modeling and analysis framework for software fault detection and correction process. *Softw. Test. Verif. Reliab.* **2016**, *26*, 351–365. [CrossRef]
34. Hu, Q.P.; Xie, M.; Ng, S.H.; Levitin, G. Robust recurrent neural network modelling for software fault detection and correction prediction. *Reliab. Eng. Syst. Saf.* **2007**, *92*, 332–340. [CrossRef]
35. Xiao, H.; Cao, M.; Peng, R. Artificial neural network based software fault detection and correction prediction models considering testing effort. *Appl. Soft Comput.* **2020**, *94*, 106491. [CrossRef]
36. Huang, C.Y.; Huang, W.C. Software reliability analysis and measurement using finite and infinite server queueing models. *IEEE Trans. Reliab.* **2008**, *57*, 192–203. [CrossRef]
37. Hwang, S.; Pham, H. Quasi-renewal time-delay fault-removal consideration in software reliability modeling. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2009**, *39*, 200–209. [CrossRef]
38. Lin, C.T. Analyzing the effect of imperfect debugging on software fault detection and correction processes via a simulation framework. *Math. Comput. Model.* **2011**, *54*, 3046–3064. [CrossRef]
39. Hu, Q.P.; Peng, R.; Ng, S.H.; Wang, H.Q. Simulation-based FDP & FCP analysis with queueing models. In Proceedings of the 2008 IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 8–11 December 2008; pp. 1577–1581.
40. Chatterjee, S.; Shukla, A. Modeling and Analysis of Software Fault Detection and Correction Process through Weibull-Type Fault Reduction Factor, Change Point and Imperfect Debugging. *Arab. J. Sci. Eng.* **2016**, *41*, 5009–5025. [CrossRef]
41. Chatterjee, S.; Shukla, A. Effect of Test Coverage and Change Point on Software Reliability Growth Based on Time Variable Fault Detection Probability. *J. Softw.* **2016**, *11*, 110–117. [CrossRef]
42. Yamada, S.; Ohba, M.; Osaki, S. S-Shaped Reliability Growth Modeling for Software Error Detection. *IEEE Trans. Reliab.* **1984**, *32*, 475–484. [CrossRef]
43. Ohba, M. Inflection S-shaped software reliability growth model. In *Stochastic Models in Reliability Theory*; Springer: Berlin/Heidelberg, Germany, 1984; pp. 144–162.
44. Yamada, S.; Tokuno, K.; Osaki, S. Imperfect debugging models with fault introduction rate for software reliability assessment. *Int. J. Syst. Sci.* **1992**, *23*, 2241–2252. [CrossRef]
45. Pham, H.; Zhang, X. An NHPP Software Reliability Model and Its Comparison. *Int. J. Reliab. Qual. Saf. Eng.* **1997**, *4*, 269–282. [CrossRef]
46. Zhang, X.; Teng, X.; Pham, H. Considering fault removal efficiency in software reliability assessment. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2003**, *33*, 114–120. [CrossRef]
47. Kapur, P.K.; Pham, H.; Anand, S.; Yadav, K. A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation. *IEEE Trans. Reliab.* **2011**, *60*, 331–340. [CrossRef]
48. Hsu, C.J.; Huang, C.Y.; Chang, J.R. Enhancing software reliability modelling and prediction through the introduction of time-variable fault reduction factor. *Appl. Math. Model.* **2011**, *35*, 506–521. [CrossRef]
49. Wood, A. Predicting software reliability. *IEEE Comput.* **1996**, *11*, 69–77. [CrossRef]
50. Pham, H.; Nordmann, L. A general imperfect-software-debugging model with S-shaped fault-detection rate. *IEEE Trans. Reliab.* **1999**, *48*, 169–175. [CrossRef]

51. Linux Kernel Bugzilla. Available online: https://bugzilla.kernel.org/ (accessed on 12 March 2016).
52. Wu, C.Y.; Huang, C.Y. A Study of Incorporation of Deep Learning into Software Reliability Modeling and Assessment. *IEEE Trans. Reliab.* **2021**, *70*, 1621–1640. [CrossRef]