International Journal of Reliability and Applications
Vol. 17, No. 2, pp. 149-158, 2016

A software reliability model with a Burr Type Ⅲ fault detection rate function

Kwang Yoon Song, In Hong Chang[*] and Min Su Choi
Department of Computer Science and Statistics, Chosun University, Gwangju, Korea

Received 21 October 2016; revised 11 December 2016; accepted 27 December 2016

Abstract: We are enjoying a very comfortable life thanks to modern civilization, however, comfort is not guaranteed to us. Development of software system is a difficult and complex process. Therefore, the main focus of software development is on improving the reliability and stability of a software system. We have become aware of the importance of developing software reliability models and have begun to develop software reliability models. NHPP software reliability models have been developed through the fault intensity rate function and the mean value functions within a controlled testing environment to estimate reliability metrics such as the number of residual faults, failure rate, and reliability of the software. In this paper, we present a new NHPP software reliability model with Burr Type Ⅲ fault detection rate, and present the goodness-of-fit of the fault detection rate software reliability model and other NHPP models based on two datasets of software testing data. The results show that the proposed model fits significantly better than other NHPP software reliability models.

Key Words: *Burr type Ⅲ function, fault detection rate, Non-homogeneous Poisson process, software reliability*

1. INTRODUCTION

We are enjoying a very comfortable life thanks to modern civilization, however, comfort is not guaranteed to us. Because the inconvenience caused by a software system failure is greater. So, software system has a great impact on our life. In recent years, software has permeated industrial equipment and consumer products. Generally, a lot of development resources are consumed by software development projects (Bokhari, 2007). Development of software system is a difficult and complex process. Therefore, the main focus of software development is on improving the reliability and stability of a software system. Furthermore, since the 1970s, we have become aware of the importance of developing software reliability models and have begun to develop software reliability models. The

[*]Corresponding Author.
E-mail address: *ihchang@chosun.ac.kr*

pioneering attempt in non-homogeneous Poisson process (NHPP) based on software reliability model was made by Goel and Okumoto (1979). Goel and Okumoto(1979) presented a stochastic model for the software failure phenomenon based on a nonhomogeneous Poisson process, and this model describes the failure observation phenomenon by an exponential curve. There are also software reliability model that describe either S-shaped curves, or a mixture of exponential and S-shaped curves. Some of the important contributions of these types of models are due to Yamada et al. (1983), Ohba (1984), Kapur and Garg (1992), Kapur et al. (1999), and Pham (2006). Once software systems are introduced, the software systems used in the field environments are the same as or close to those used in the development-testing environment. Most of NHPP software reliability model were developed based on the assumptions that faults detected in the testing phase are removed immediately with no debugging time delay, and no new fault introduction. And also, NHPP software reliability models have been developed through the fault intensity rate function and the mean value functions $m(t)$ within a controlled testing environment to estimate reliability metrics such as the number of residual faults, failure rate, and reliability of the software.

Burr (1942) was proposed several types of the cumulative functions and has stressed the advantages obtained by the direct use of the cumulative function. A number of useful functions have been considered. A general method for fitting any cumulative function by the construction of a table has been suggested. A particular method depending upon the use of certain new cumulative moments has been given. Nahed (2005) is discussed the reliability of a system when both the strength of the system and the stress imposed on it are independent, non-identical Burr Type Ⅲ distributed random variables.

Ahmad et al. (2011) developed a software reliability growth model based on the non-homogeneous Poisson process which incorporates the Burr Type Ⅲ testing-effort. This scheme has a flexible structure and may cover many of the earlier results on software reliability growth modeling.

In this paper, we discuss a new NHPP software reliability model with different fault detection rate. We examine the goodness-of-fit of the fault detection rate software reliability model and other NHPP models based on two datasets of software testing data. The explicit solution of the mean value function for the new NHPP software reliability model is derived in Section 2. Criteria for model comparisons and selection of the best model are discussed, and model analysis and results are discussed in Section 3. Section 4 presents the conclusions and remarks.

2. A NEW NHPP SOFTWARE RELIABILITY MODEL

2.1 Related research

The NHPP models provide an analytical framework for describing the software failure phenomenon during testing. The main point in the NHPP models are to estimate the mean value function (MVF) of the cumulative number of failures experienced up to a certain point in time. The NHPP models will have with different functional forms of the mean value function with each different assumption. The software fault detection process has been widely formulated by using a counting process. A counting process $\{N(t), t \geq 0\}$, is said to be a NHPP with intensity function $\lambda(t)$, if $N(t)$ follows a Poisson distribution with the mean value function $m(t)$, i.e.,

$$\Pr\{N(t) = n\} = \frac{\{m(t)\}^n}{n!}\exp\{-m(t)\}, n = 0,1,2,3 \ldots.$$

The mean value function $m(t)$, which is the expected number of faults detected at time t with $m(0) = 0$ can be expressed as

$$m(t) = \int_0^t \lambda(s)ds.$$

The software reliability $R(x|t)$ is defined as the probability that a failure does not occur in the time interval $[t, t + x](t \geq 0, x \geq 0)$

$$R(x|t) = e^{-[m(t+x)-m(t)]}.$$

Many NHPP-based SRGM have been modeled $m(t)$ using the differential equation

$$\frac{d\,m(t)}{dt} = b(t)[a - m(t)] \tag{1}$$

Solving Eq. (1) makes it possible to obtain different values of $m(t)$ using different values for $b(t)$, which reflects various assumptions of the software testing process. The solution for the mean value function m(t), where the initial condition m(0) = 0, is given

$$m(t) = a(1 - e^{-\int_0^t b(s)ds}) \tag{2}$$

(1) Goel-Okumoto model (Goel and Okumoto, 1979)

Goel-Okumoto model is an exponential NHPP model. For $b(t) = b$,

$$m(t) = a(1 - e^{-bt}).$$

(2) Delayed S-shaped model (Yamada et al, 1983)

Delayed S-shaped model is a S-shaped NHPP model. For $b(t) = \frac{b^2t}{bt+1}$,

$$m(t) = a(1 - (1 + bt)e^{-bt}).$$

(3) Inflection S-shaped model (Ohba, 1984)

Inflection S-shaped model is also a S-shaped NHPP model. For $b(t) = \frac{b}{1+\beta e^{-bt}}$,

$$m(t) = \frac{a(1-e^{-bt})}{1+\beta e^{-bt}}.$$

where $m(t)$ is mean value function in the NHPP model, $b(t)$ is fault detection rate, a is expected total number of faults that exist in the before testing, β is the inflection factor.

2.2. A Burr Type Ⅲ fault detection rate model

We propose an NHPP software reliability model with different fault detection rate using Eq. (2) and the following assumptions:

(a) The occurrence of software failures follows an NHPP.
(b) Software can fail during execution, caused by faults in the software.
(c) The software-failure detection rate at any time is proportional to the number of remaining faults in the software at that time.
(d)The fault detection rate differs among faults.

(e) When a software failure occurs, a debugging effort removes the faults immediately, and no new errors are introduced.
(f) The curve of function is described by Burr Type Ⅲ fault detection rate function.

In this paper, we consider a Burr Type Ⅲ fault detection rate function $b(t)$ to be as follows:

$$b(t) = \frac{bkt^{bk-1}(1-t^k)^{-b-1}}{1-(1+t^{-k})^{-b}},\ b, k > 0 \tag{3}$$

where b and k are unknown parameters. Figure 1 show the graph of the Burr type Ⅲ fault detection rate function $b(t)$.

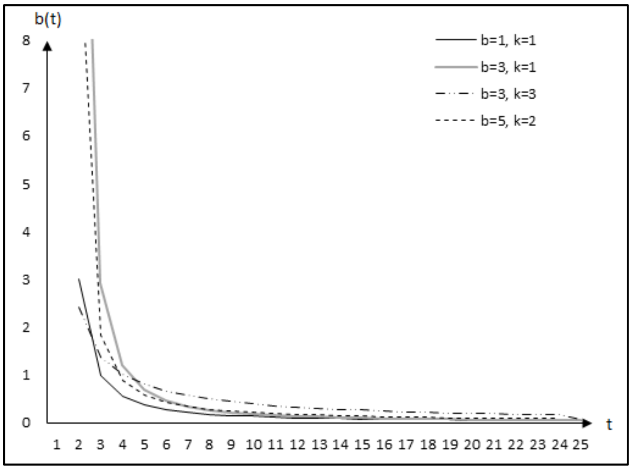

Figure 1. Fault detection rate function b(t)

We obtain a new NHPP software reliability model with a Burr type Ⅲ fault detection rate function, m(t), that can be used to determine the expected number of software failures detected by time t by substituting the function b(t) above into Eq. (2):

$$m(t) = a\left(1 - e^{-\left(b\ln(t^k+1) - \ln\left((t^k+1)^b - t^{bk}\right)\right)}\right) \tag{4}$$

Table 1. NHPP Software reliability models

No.	Model	$m(t)$
1	G-O model	$m(t) = a(1 - e^{-bt})$
2	Delayed S-shaped model	$m(t) = a(1 - (1 + bt)e^{-bt})$
3	Inflection S-shaped model	$m(t) = \frac{a(1 - e^{-bt})}{1 + \beta e^{-bt}}$
4	Proposed New Model	$m(t) = a\left(1 - e^{-\left(b\ln(t^k+1) - \ln\left((t^k+1)^b - t^{bk}\right)\right)}\right)$

Table 1 summarizes the proposed new model and several existing well-known NHPP models with different mean value functions.

3. NUMERICAL EXAMPLES

Once the analytical expression for the mean value function $m(t)$ is derived, the model parameters to be estimated in the mean value function can then be obtained with the help of a developed MATLAB and EXCEL 2010 program based on the least-squares estimate (LSE) method.

3.1 Criteria for model comparisons

Criteria for model comparisons will be used as criteria for the model estimation of the goodness-of-fit and to compare the proposed model and other existing models as listed in Table 1. For all criteria (Pham, 2006) the smaller the value, the closer the model fits relative to other models run on the same data set. The mean squared error (MSE) measures the distance of a model estimate from the actual data with the consideration of the number of observations, n, and the number of unknown parameters in the model, m. The mean squared error is given by

$$\mathrm{MSE} = \frac{\sum_{i=0}^{n}(m(t_i)-y_i)^2}{n-m}$$

where $m(t_i)$ is the estimated cumulative number of failures at t_i for $i = 1,2,\cdots,n$; and y_i is the total number of failures observed at time t_i.

The sum absolute error is similar to the sum squared error, but the way of measuring the deviation is by the use of absolute values, and sums the absolute value of the deviation between the actual data and the estimated curve. The sum absolute error is given by

$$\mathrm{SAE} = \sum_{i=0}^{n}|m(t_i) - y_i|.$$

Akaike's information criterion (Akaike, 1974) is given by

$$\mathrm{AIC} = -2\log|\mathrm{MLF}| + 2m$$

where m is the number of unknown parameters in the model.

The predicted sum squared error sum the value of the deviation between the actual data and the estimated value after the point at which the parameters are estimated. The predicted sum squared error is given by

$$\mathrm{PreSSE} = \sum_{i=t_i+1}^{n}(m(t_i) - y_i)^2.$$

3.2 Data information

Dataset1 (DS1) and Dataset2 (DS2), field failure data listed in Table 2, was reported by Lee et al. (1998). The field failure data is the failure data detected in the system test. The size of the exchange software is a large program with 134 million source code lines and consists of 140 major functional blocks. All faults detected for each system test are registered in the fault management system (FMS) and are tracked until all faults have been corrected and solved.

Table 2. Datasets

Time Index (Month)	DS1		DS2	
	Failures	Cumulative Failures	Failures	Cumulative Failures
1	23	23	83	83
2	21	44	287	370
3	81	125	177	547
4	24	149	193	740
5	22	171	120	760
6	12	183	67	927
7	2	185	75	1002
8	1	186	46	1048
9	0	186	24	1072
10	11	197	69	1141
11	16	213	129	1270
12	30	243	117	1387
13	2	245	31	1418
14	0	245	40	1458
15	1	246	34	1492
16	0	246	35	1527
17	2	248	20	1547
18	0	248	5	1552

3.3 Results

Table 3 and Table 4 summarize the results of the estimated parameters of all 4 models in Table 1 using the least-squares estimation (LSE) technique. We obtained the three common criteria when $t = 1,2,\cdots,18$ from DS1, as can be seen from Table 3, MSE, SAE, and AIC value for the proposed new model are the lowest values compared to all models. And, we obtained the MSE when $t = 1,2,\cdots,13$ from DS2, and obtained the PreSSE when $t = 14,\cdots,18$, as can be seen from Table 4, MSE and PreSSE value for the proposed new model are the lowest values compared to all models. As a result, the proposed new model has better fitting than the other models.
Figure 2 and 5 show the graph of the mean value functions for all models for DS1 and DS2, respectively. Figure 3 shows the graph of the absolute value of relative error for all models, and better when close to 0 at each point. Figure 4 shows the graph of the reliability for all models for DS1. Figure 6 shows the graph of the predicted values of mean value function for all models.

Table 3. Model parameter estimation and comparison criteria from DS1

Model	LSE's	MSE	SAE	AIC
GO Model	$\hat{a}$=261.587, $\hat{b}$=0.175	250.077	275.687	217.486
Delayed S-shaped Model	$\hat{a}$=237.331, $\hat{b}$=0.464	317.048	285.250	264.427
Inflection S-shaped Model	$\hat{a}$=261.093, $\hat{b}$=0.177, $\hat{\beta}$=0.011	266.735	277.546	217.271
Proposed New Model	$\hat{a}$=294.171, $\hat{b}$=3.768, $\hat{k}$=1.073	197.251	255.176	191.191

Table 4. Model parameter estimation and comparison criteria from DS2

Model	LSE's	MSE	PreSSE
GO Model	$\hat{a}$=1779.975, $\hat{b}$=0.115	3506.792	3250.215
Delayed S-shaped Model	$\hat{a}$=1341.024, $\hat{b}$=0.0.409	7333.963	185415.396
Inflection S-shaped Model	$\hat{a}$=1779.982, $\hat{b}$=0.0.115, $\hat{\beta}$=0.00001	3857.476	3249.519
Proposed New Model	$\hat{a}$=3010.569, $\hat{b}$=4.300, $\hat{k}$=0.628	2955.549	3216.632

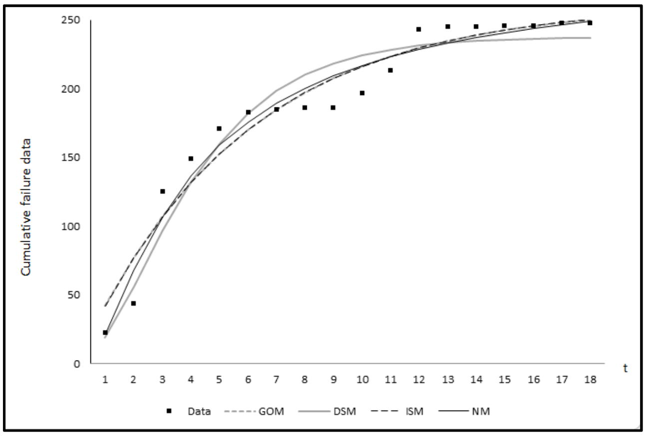

Figure 2. Mean value function of all models for DS1

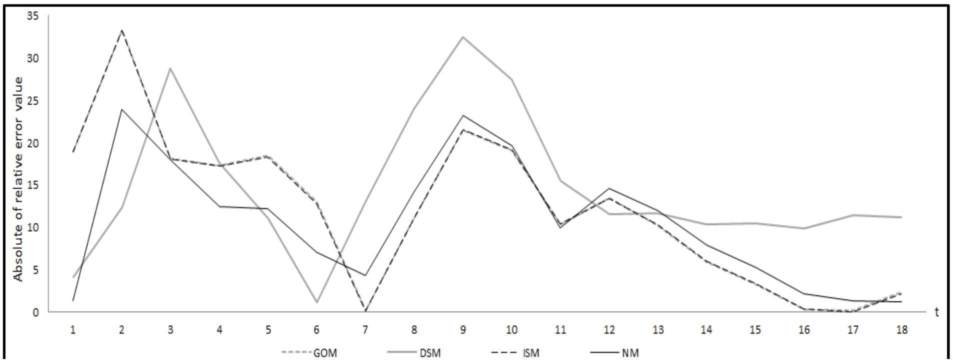

Figure 3. Absolute of relative error value of models for DS1

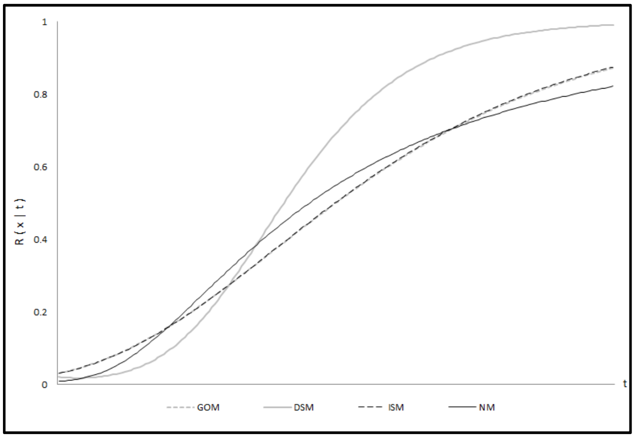

Figure 4. Reliability of all models for DS1

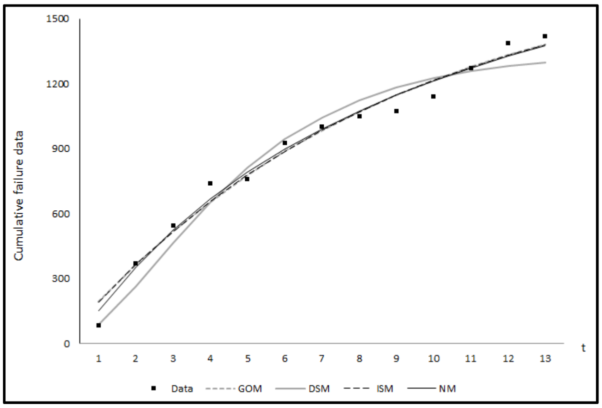

Figure 5. Mean value function of all models for DS2

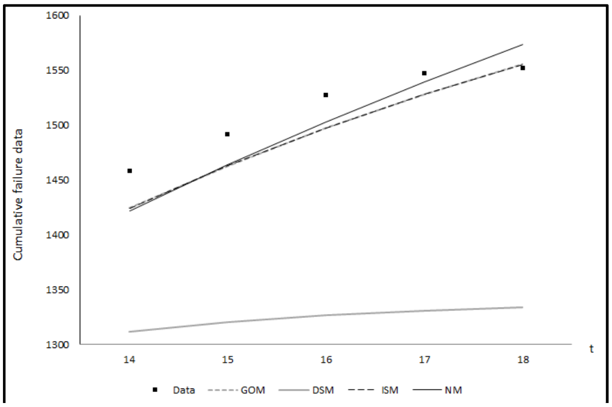

Figure 6. Predicted values of mean value function of all models for DS2

4. CONCLUSIONS

Most of NHPP software reliability models are applied to software testing data and then used to make predictions on the software failures and reliability in the field. In this paper, we discussed a new NHPP software reliability model with different fault detection rate. We examined the goodness-of-fit of the fault detection rate software reliability model and other NHPP models based on two datasets of software testing data. Table 3 and Table 4 summarize the results of the estimated parameters of all 4 models in Table 1 using the least-squares estimation (LSE) technique. As can be seen from Table 3, MSE, SAE, and AIC value for the proposed new model are the lowest values compared to all models. And also, as can be seen from Table 4, MSE and PreSSE value for the proposed new model are the lowest values compared to all models. As a result, the proposed new model has better fitting than the other models. In order to accurately evaluate the reliability, further study on the parameters to be applied to each model in a multifaceted aspect is needed.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2009277 and NRF-2015R1D1A1A01060050).

REFERENCES

Ahmad, N., Quadri, S. M. K., Khan, M. G. M. and Kumar, M. (2011). Software reliability growth models incorporating burr type Ⅲ test-effort and cost-reliability analysis, *International Journal of Computer Science and Information Technologies*, **2**, 555-562.

Akaike, H. (1974). A new look at statistical model identification, *IEEE Transactions on Automatic Control*, **19**, 716-719.

Bokhari, M. U. and Ahmad, N. (2007). Software reliability growth modeling for exponentiated weibull function with actual software failures data, *Advances in Computer Science and Engineering: Reports and Monographs*, World Scientific Publishing Company, Singapore, **2**, 390-396.

Burr, I. W. (1942). Cumulative frequency functions, *The Annals of Mathematical Statistics*, **13**, 215-232.

Goel, A. L. and Okumoto, K. (1979). Time dependent error detection rate model for software reliability and other performance measures, *IEEE Transactions on Reliability*, **28**, 206-211.

Kapur, P. K., Garg, R. B. (1992). A software reliability growth model for an error removal phenomenon, *Software Engineering Journal*, **7**, 291-294.

Kapur, P. K., Garg, R. B., Kumar, S. (1999). *Contributions to hardware and software reliability*, Singapore: World Scientific Publishing Co. Ltd.

Lee, J. K., Shin, S. K. and Lee, Y. M. (1998). A Study on Hypothetical Switching Software Through of the Analysis of Failure Data, *The Journal of Korean Institute of Communications and Information Sciences*, **23**, 1915-1925.

Mokhlis, N. A. (2005). Reliability of a stress-strength model with burr type Ⅲ distributions, *Communications in Statistics – Theory and Methods*, **34**, 1643-1657.

Ohba, M. (1984). Inflexion S-shaped software reliability growth models, *Stochastic Models in Reliability Theory*, Osaki, S. and Hatoyama, Y. (eds.), Springer-Verlag, Berlin, 144-162.

Pham, H. (2006). *System Software Reliability*, Springer, London, 149-149.

Yamada, S., Ohba, M. and Osaki, S. (1983). S-shaped reliability growth modeling for software fault detection, *IEEE Transactions on Reliability*, **32**, 475-484.

Yamada, S. and Osaki, S. (1985). Software reliability growth modeling: models and applications, *IEEE Transaction on Software Engineering*, **11**, 1431-1437.