Indian Journal of Science and Technology, *Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/101785, October 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

The Performance Analysis of the Software Reliability NHPP Log-linear Model Depend on Viewpoint of the Learning Effects

Kyung-Soo Kim[1] and Hee-Cheul Kim[2]*

[1] Department of Internet Information, Baek Seok Culturel University, South Korea; kkskim@bscu.ac.kr
[2]Departmentof Industrial and Management Engineering, Namseoul University, South Korea; kim1458@nsu.ac.kr

Abstract

Objectives: In the procedure of the developing software product, the software administrators are needed to have tools/ mechanisms to detect software failures. In this research, the test implements with perspective about actual learning effects were considered based on the software reliability models. **Methods/Statistical Analysis**: The log-linear pattern as the lifetime distribution was offered using the finite failure NHPP. Software error detection techniques contain self-directed errors-noticed element and learning element. These two elements can be discussed from the influencing elements. When applied to log-linear property model, the learning element was more efficient than the self-directed errors-noticed element model. **Findings**: When applied to log-linear property model, the learning element was more efficient than the self-directed errors-noticed element model. The examination of a failure time data considering the influencing elements was achieved using software failure interval time and the maximum likelihood estimation was used for parameter estimation. Besides, the competence of the data from the trend analysis was established. And, the model choice was skilled using MSE and R^2. The reliability considering the influential elements for each model is showing a decrease pattern. In terms of reliability, the greater learning element has revealed the low reliability. In terms of judgment of the reliability, a case of the reliability for the assumed mission time, the self-directed errors-noticed element than the learning element model has shown the lower reliability. Namely, the cases of the learning element are greater or equal greater than self-directed errors-noticed element shows propensity to rise slightly, but a property of the reliability shows tend to rise progressively have the decreasing pattern for the mission time. Eventually, the reliability has sensitive property for the learning element about the mission time. **Improvements/Applications**: The software testing for the debugging to reduce cost in terms of the reliability from software is an essential problem. From a research, the software developers must be considered for the growth model by the prior knowledge of the software to identify failure modes which can be able to help.

Keywords: Learning Effects, Log-linear Distribution, Mission Time, Non-Homogeneous Poisson Process, Trend Analysis

1. Introduction

So far, the various software reliability models have been suggested. This model depends on Non-Homogenous Poisson Process (NHPP) was acknowledged to an outstanding model[1,2] through the error discovery progression. Also, this model has norms that if an error occurs, the fault was removed instantaneously through the debugging procedure and no new error has occurred. A further refine the model based on the Enhanced Non-Homogenous Poisson Process (ENHPP) was accomplished by[1]. As well,[2] were proposed for an exponential software reliability forms. For this model, the whole number of the faults based on the S-shaped or exponential-formed property was used for the mean value function. The delayed s-formed reliability progress model and inflection s-formed reliability progress model based on these models were proposed by in[3,4] was proposed about software reliability difficulties from the changing point and[5] was proposed the widespread reliability evolution models. In connection with the model[6] were proposed the problem that testing measured coverage for the equilibrium of model with software steadiness can be

**Author for correspondence*

assessed.Relatively[7] was proposed the problem that the general logistic testing-exertion property and the changing point factor by the integrating effectual methods can be forecasted the software reliability. In[8] can be explained about the learning element that the software managers to become accustomed with the software situation and test implements for s-formed model. In this study, the characteristics for the log-linear distribution depends on finite failure NHPP were compared. The software reliability model considering the significant factors using the self-directed errors-noticed feature and learning element was planned.

2. Basic Work

2.1 Finite NHPP Model using GOS

In a course of time part[9], software reliability classics have the hypothesis that software failures exhibition the performance of a NHPP. The element $\lambda(t)$ denotes the failure intensity property of the software at period t and time-suspended from the stochastic procedure. $N(t)$ follows a Poisson PDF (probability density function) with stricture $m(t)$. In other words, $m(t) = E[N(t)]$ is connected as follows:

$$m(t) = \int_0^t \lambda(s)ds \tag{1}$$

And,

$$\frac{dm(\mathrm{t})}{dt} = \lambda(\mathrm{t}) \tag{2}$$

In this occasion, the Poisson PDF is formed as follows:

$$p(N(t) = n) = \frac{[m(t)]^n}{n!} e^{-m(t)}, \; n = 0, 1, \cdots, \infty \tag{3}$$

The several period field prototypes can be performed the stochastic failure procedure by NHPP. These models have relation from the failure intensity property $\lambda(t)$ and mean value property $m(t)$ .The NHPP models contain the finite failure characteristic and infinite failure property. The finite failure NHPP models have the norms that the anticipated number of the errors was perceived throughout the infinite measure of the testing period will be limited. Thus, the prototypical form using GOS (General Order Statistics) follows finite failure NHPP. The displaying value θ with the expected property of the errors can be perceived finite failure HPP models. From $F(t)$ (cumulative distribution function) and $m(t)$ (mean value function), the finite failure NHPP model was acknowledged next form.

$$m(t) = \theta \; F(t) \tag{4}$$

In the Equation (4), the failure intensity form $\lambda(t)$ for the finite failure NHPP prototype can be expressed next condition.

$$\lambda(t) = \theta \; F'(t) = \theta \, f(t) \tag{5}$$

The Equation (5) can be transformed next form.

$$\frac{\lambda(t)}{[\theta - m(t)]} = h(t) \tag{6}$$

In the Equation (6), $h(t)$ (refer to hazard function) means the failure occurrence percentage per error of the software through out testing. The amount $[\theta - m(t)]$ denotes the expected amount of the errors remaining in the software at time t. A characteristic of the failure occurrence percentage per error $h(t)$ has a constant, growing and reducing. If $\{t_n, n = 1, 2, \cdots\}$ indicates the arrangement about the failure interval time, the failure time t_n indicates the time about between $(n-1)^{st}$ and n^{th} failure. Thus, the last failure time x_n may be embodied in the following expression.

$$x_n = \sum_{i=1}^{n} t_i \tag{7}$$

The likelihood function of $x_1, x_2, \ldots, x_n$ was known to the following expression[10].

$$f_{X_1, X_2, \cdots, X_n}(x_1, x_2, \ldots, x_n) = e^{-m(x_n)} \prod_{i=1}^{n} \lambda(x_i) \tag{8}$$

If the construction about failure times $(x_1, x_2, \ldots, x_n)$ from observations for the random variables $(X_1, X_2, \ldots, X_n)$ is given, the parameter estimation for the reliability models can be realized from method of the maximum likelihood method (MLE). Further more, using the failure last time x_n and the mission time ξ, the conditional reliability $\hat{R}(\xi \mid x_n)$ was known as follows[11]:

$$\hat{R}(\xi \mid x_n) = e^{-\int_{x_n}^{x_n+\xi} \lambda(\tau)d\tau} = \exp[-\{m(\xi + x_n) - m(x_n)\}] \tag{9}$$

3. Distribution Function and Probability Density Function Considering Influential Factor

Software testing procedure using the learning properties is an important process. These jobs by the management can be the equivalent or manipulation of these effects. The possible action from this work was reflected the software reliability in any mode. The influential features contain the self-directed errors-noticed element γ and learning feature η. The influential elements can be perceived for the outcome of software errors. In other words, the $f(t)$ means probability density function that denotes the segment of the errors noticed at time t and $F(t)$ denotes the cumulate distribution function that denotes the piece of the errors noticed within time $(0,t]$. Thus, $(1-F(t))$ was regarded as the segment of the errors as yet unnoticed at time t. The model, in view of the influence elements, can be employed next condition[12].

$$f(t)=(\gamma+\eta F(t))\,(1-F(t)) \tag{10}$$

From equation (10), $\gamma>0$ and $\eta>0$ represent parameters.

The self-directed errors-noticed element means component spontaneously for the outcome of software errors which the testing software developer staffs were unnoticed. In contrast, the learning element means that the interesting software developer staffs purposefully set out for the discovering of software errors in software structure using the failure forms which were previously perceived. Using the both elements, the efficiency of a software debugging can be improved. Using the Equation (10), the following hazard function can be modified.

$$h(t)=\frac{f(t)}{1-F(t)}=(\gamma+\eta F(T)) \tag{11}$$

Using the Equation (11), if the cumulate distribution function and probability density function are reflected the self-directed errors-noticed element and learning element, it can be embodied as next form.

$$F(t)=\frac{h(t)-\gamma}{\eta}\quad,\ f(t)=F'(t)=\frac{h'(t)-\gamma}{\eta} \tag{12}$$

4. Log-liner Property Software Reliability NHPP Model using the Viewpoint of the Learning Properties

In this section, the log-linear model was applied. Thus, the hazard function was known to next form[13].

$$h(t)=\frac{F'(t)}{1-F(t)}=\frac{f(t)}{1-F(t)}=e^{\alpha+\beta t} \tag{13}$$

In the Equation (13), α is intercept and β denotes slope parameter.

Using the Equation (13) and (14), if the cumulate distribution function and probability density function can be transmitted the influence element; it can be embodied in the following expression.

$$F(t)=\frac{h(t)-\gamma}{\eta}=\left[\frac{e^{\alpha+\beta t}-\gamma}{\eta}\right],\ f(t)=F'(t)=\frac{h'(t)-\gamma}{\eta}=\left[\frac{\beta e^{\alpha+\beta t}}{\eta}\right] \tag{14}$$

Using the Equation (4) and (5), the mean value function and intensity function for the finite failure NHPP model are reflected the knowledge effect; it can be conveyed in the following expression.

$$m(t)=\theta\ F(t)=\theta\left[\frac{e^{\alpha+\beta t}-\gamma}{\eta}\right],\quad \theta>0,\ \alpha,\beta>0,\ t\geq 0.$$

$$\lambda(t)=\theta\ F'(t)=\theta\left[\frac{\beta e^{\alpha+\beta t}}{\eta}\right] \tag{15}$$

In this situation, the likelihood function using the Equation (15) and (9) can be implemented as follows.

$$L_{NHPP}(\alpha,\beta\mid \underline{x})=\prod_{i=1}^{n}\left(\theta\frac{\beta e^{\alpha+\beta x_i}}{\eta}\right)\exp\left(-\theta\frac{e^{\alpha+\beta x_n}-\gamma}{\eta}\right) \tag{16}$$

Note that $\underline{x}$ means arrangement of software failure periods $(0,x_1,x_2,x_3\cdots,x_n]$.

In this situation, the intercept α was assumed to be zero to ease the parameter estimation.

For the parameter estimation, the log likelihood function can be derived as next form using the Equation (16).

$$\ln L_{NHPP}(\beta\mid \underline{x})=n\ln\theta+n\ln\beta-n\ln\eta+\beta\sum_{i=1}^{n}x_i-\frac{\theta}{\eta}\left(e^{\beta x_n}-\gamma\right) \tag{17}$$

So, using the maximum likelihood estimation, $\hat{\theta}_{MLE}$ and $\hat{\beta}_{MLE}$ can be estimated.

$$\frac{\partial \ln L_{NHPP}(\beta\mid \underline{x})}{\partial\theta}=\frac{n}{\theta}-\frac{1}{\eta}\left(e^{\beta x_n}-\gamma\right)=0 \tag{18}$$

$$\frac{\partial \ln L_{NHPP}(\beta\mid \underline{x})}{\partial\beta}=\frac{n}{\beta}+\sum_{i=1}^{n}x_i-\frac{\theta}{\eta}\beta e^{\beta x_n}=0 \tag{19}$$

Note that $\underline{x}$ indicates arrangement of software failure times $(0,x_1,x_2,x_3\cdots,x_n]$.

5. Model Selection using Real Dataset

To compare the efficiency of the proposed model, we used to the comparison criteria[14]. The criteria for the efficiency can be used the mean square error and R square in this field. The deviations among the forecasted values and the real realizations measure the mean square error (MSE) and it is well-defined as next form.

$$MSE = \frac{\sum_{i=1}^{n}[m(x_i) - \hat{m}(x_i)]^2}{n-k} \qquad (20)$$

In the Equation (20), $m(t_i)$ means the whole cumulated amount of the errors noticed within period is $(0, t_i]$ and $\hat{m}(t_i)$ indicates the estimated cumulative amount of errors at period t_i. The estimated accumulative amount of the errors at time was gained from the appropriating mean value function. Also, n is the amount of realizations and k shows the amount of parameters. That is, the model with the smaller mean square error values can be reflected a proficient model. Further more, the R square (R^2) can be measured how successful fit in aspect of the explaining power for the variance from the assessed data. It is well-defined as next form consequently, the model with the larger R^2value is considered as the more efficient model.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}[m(x_i) - \hat{m}(x_i)]^2}{\sum_{i=1}^{n}\left(m(x_i) - \sum_{j=1}^{n} m(x_j)/n\right)^2} \qquad (21)$$

6. Conclusion Software Failure Data Analysis

In this section, the failure time information[15]aboutthe software failures were used to study the property for the learning features. Table 1 is failure time information.

Table 1. Software failure time information

Failure Number	*Failure Time (hours)*	Failure Number	*Failure Time (hours)*
1	0.0094	16	10.0192
2	0.0500	17	10.4077
3	0.4064	18	10.4791
4	4.6307	19	11.0706
5	5.1741	20	11.3250
6	5.8808	21	11.5284
7	6.3348	22	11.9226
8	7.1654	23	12.0294
9	7.2316	24	12.0740
10	8.2604	25	12.1835
11	9.2962	26	12.3549
12	9.3812	27	12.5381
13	9.5223	28	12.8049
14	9.8783	29	13.4615
15	9.9346	30	13.8530

The Laplace trend test[16], in the first, should be preceded to determine the effectiveness of the data. The outcomes of this test in this Figure 1display that value of the Laplace factors have the shape from 2 and –2. Because the properties of the reliability growth show, it performs to be the dangerous value is not existed. Consequently, an approximation of the reliability may be possibleusing this data[16]. The maximum likelihood method was performed to estimate parameters for each model. The convert variable data($Failure\ time \times 10^{-3}$) from the original failure time data was applied to facilitate parameter estimation. A result of the parameter estimation was listed in Table 2.These calculations were achieved iteration of 100 times, solving numerically, using the initial value from 0.0001to5.0. For the checking acceptable convergent, the tolerance value for width of the interval value used to 10^{-5} using C-language. The estimated value of the Mean Square Error (MSE) and R-square (R^2) were listed in Table 2. In this table, the case of the generally having the greater self–directed errors-noticed element than learning element is efficient using the mean square error. But, in terms of the coefficient of determination, the learning elements are the higher property with the increasing explanatory power. Thus, the higher learning

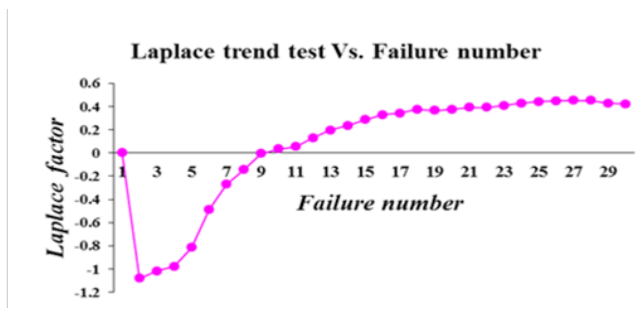
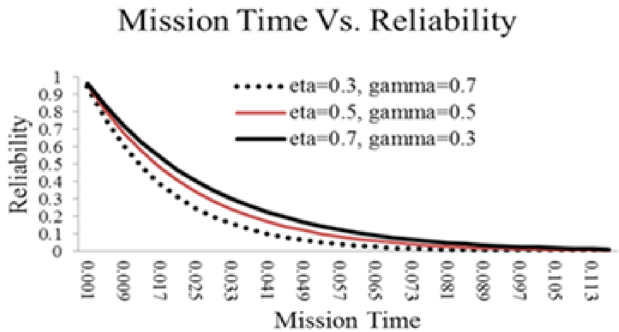


Figure 1. Test of Laplace trend.

Table 2. MLE , MSE and R^2 considering influential factors for each model

Influential features		*Log-linear model*			
η	γ	$\hat{\theta}_{MLE}$	$\hat{\beta}_{MLE}$	MSE	R^2
0.3	0.7	29.6723	0.5539	**8603.590**	0.8575
0.5	0.5	29.4123	0.7129	8299.678	0.8577
0.7	0.3	29.3066	0.8419	8170.015	**0.8579**

Note. In the Table 2, η denotes learning feature, γ is the self-directed errors-noticed element and MLE means maximum likelihood estimation. MSE indicates the mean square error. Also,R^2 is the coefficient of determination. In this table, the smallest and highest estimation value was indicated in boldface.



Note. In the figure above, eta is η and gamma is γ.

Figure 2. Comparison reliability considering influential factors for each model.

element model is more useful model. In the Figure 2, the reliability considering the influential elements for each model is showing a decrease pattern. In terms of reliability, the greater learning element as revealed the low reliability. In terms of judgment of the reliability, a case of the reliability for the assumed mission time, the self-directed errors-noticed element than the learning element model has shown the lower reliability. Namely, the cases of the learning element are greater or equal greater than self–directed errors-noticed element shows propensity to rise slightly, but a property of the reliability show send to rise progressively have the decreasing pattern for the mission time. Eventually, the reliability has sensitive property for the learning element about the mission time. Therefore, the log-linear model was judged more reliable model in this field.

7. Conclusion

The features of the log-linear model, constructed on finite failure NHPP for influential elements considering the self-directed errors-noticed element and the learning element, were considered in this research. In this research, the resulting conclusions were accomplished. In term of the influential elements, the log-linear model is generally having the self-directed errors-noticed element is greater than the learning element is efficient. Commonly, when the learning element is the highest and the self-directed errors-noticed element is the lowest, the model was achieved as the effective model. From the coefficient of determination, the higher learning element model is more utility model. In terms of the reliability, the greater learning element has shown the higher reliability. Eventually, the reliability has sensitive property for the learning element about the mission time. Hence, in this research, the features for the log-linear model can be regarded as a substitute reliability model in this ground. Are placed study for the software reliability will be accomplished.

8. References

1. Gokhale SS, Trivedi KS. A time/structure based software reliability model. Annals of Software Engineering. 1999 Feb; 8(1):85–121.
2. Goel AL, Okumoto K. Time dependent error-detection rate model for software reliability and other performance measure. IEEE Transaction on Reliability. 1979 Aug; R-28(3):206–11.
3. Yamada S, Ohba H. S-shaped software reliability modeling for software error detection. IEEE Transaction on Reliability. 1983 Dec; R-32(5):475–84.
4. Zhao M. Change-point problems in software and hardware reliability. Communication in Statistics Theory and Methods. 1993 Jan; 22(3):757–68.
5. Shyur HJ. A stochastic software reliability model with imperfectde bugging and change-point. Journal of System Software. 2003 May; 66(2):135–41.
6. Pham H, Zhang X. NHPP software reliability and cost model swith testing coverage. European Journal of Operational Research. 2003 Mar; 145(2):443–54.
7. Huang CY. Performance analysis of software reliability Ngrowth models with testing-effort and change-point. JoNurnal of System Software. 2005 May; 76(2):181–94.
8. Kuei-Chen C, Yeu-Shiang H, Tzai-Zang L. A study of software reliability growth from the perspective of learning effects. Reliability Engineering and System Safety. 2008 Oct; 93(10):1410–21.
9. Kim HC, Kim JB. The comparative study of generalized goel-NHPP model using software reliability cost model. International Journal of Applied Engineering Research. 2014;9(24):25471–7.
10. Kuo L, Yang TY. Bayesian computation of software reliability. Journal of Computational and Graphical Statistics.1995 Mar;4(1):65–82.
11. Kim HC, Kim KS. The finite NHPP software cost model considering half-logistic distribution. International Journal of Applied Engineering Research. 2014 Jan; 9(22):13893–14902.
12. Kim HC**.** The comparative study of NHPP half-logistic distribution software reliability model using the perspective of learning effects. Journal of Next Generation Information Technology. 2013 Oct; 4(8):132–9.

13. RaYS, Kim HC. Study for predict of the future software failure time using nonlinear regression. IT Convergence and Security 2012. 2013; 215:1117–24.
14. Kim HC. The property of learning effect based on delayed software s-shaped reliability model using finite NHPP software cost model. Indian Journal of Science and Technology.2015Dec; 8(34):1– 7.
15. Sylwia KB. Makeham's generalized distribution. Computational Methods in Science and Technology. 2007 Sep; 13(2):113–20.
16. You TH. The infinite NHPP software reliability model based on monotonic intensity function. Indian Journal of Science and Technology. 2015Jul; 8(14):1–7.