

A Study on Comparative Evaluation of Software Reliability Model According to Learning Effect of Exponential-exponential Distribution

Hee-Cheul Kim^{1*}

¹Associate Professor, Department of Industrial & Management Engineering, Namseoul University, South Korea

*Corresponding Author

ORCID: 0000-0002-4786-5672

Abstract

In this study, the software managers were studied the NHPP software failure model using the efficient learning factor technique for software test tools during the development and testing of software products. The life distribution using an exponential-exponential distribution was based on finite failure NHPP. The software error detection technique was composed autonomous errors-detected factor that do not know in advance that was founded automatically and learning effect, that the factors were set up by the testing administrator to detect errors in detail by prior knowledge. As a result, the model that the learning factor is greater than autonomous errors-detected factor is generally efficient model could be confirmed. In this study, the result of the mean value function pattern reflecting learning effect was show a non-decreasing form. If the model following greater the learning factor is than model following autonomous errors-detected factor, the estimated value error difference in comparison with the true value could be confirmed smaller. The overall mean square error is smaller when the learning factor is larger than the autonomous errors-detected factor, so it can be regarded as an efficient model in terms of a measure of accuracy. Also, the larger learning factor following was the larger the estimated value of the coefficient of determination. Since all the proposed models in terms of coefficient of determination reflected on learning effect have more than 50%, all models are judged to be efficient models. So, the larger learning factor than autonomous errors-detected factor can be regarded as an efficient model in terms of goodness-of-fit. In the form of the reliability function, as the mission time passes, it gradually appears as a non-increasing pattern. And, in case of the larger learning factor than the autonomous errors-detected factor, the higher pattern shows in terms of the reliability. This study suggests that software managers can use the exponential-exponential learning factor as a basic guideline to detect software failures.

Keywords: Non-homogeneous Poisson process, Exponential-exponential distribution, Autonomous errors-detected factor, Learning factor, Mission time

I. INTRODUCTION

Software systems can be regarded as basic elements of significant intellectual factors of industrial management. Such a system can provide high-quality service to software users if it

can accomplish correctness and reliability. Therefore, software maintenance is to support the accuracy and reliability of the software system. The software reliability growth model, which is an engineering study related to such maintenance, has been presented. These studies were developed a software reliability model [1] that follows the non-homogeneous Poisson process (NHPP), which can be predicted the failure intensity function and the mean value function using the reliability attribute factors such as the number of remaining failures. Software reliability models are used to predict and test software reliability, number of failures remaining, failure intensity, and total cost of software development. Under these circumstances, Yamada and Osaki [2] were emphasized that the results of the mean value function can be predicted using the maximum likelihood estimation method. Also, the reliability characteristics of the mean value function was can be explained using the graph of the confidence interval of the mean value function [1, 3]. The defect detection rate developed using the exponential distribution, which is the basic model in this field, consists of the intensity function which is a constant (hazard function) [4, 5]. And Kim [6] also was studied the reliability of life distribution using Burr-XII and Type-2 Gumbel distributions.

In this study, was presented the problem of the characteristics of the learning effect that was established by the testing managers to detect errors in detail by prior knowledge.

II. RELATED RESEARCH

II.I Exponential-exponential distribution

Among the models widely used in the field of software reliability, a special form of the Weibull exponential distribution is the exponential-exponential distribution. The probability density function (PDF) and the cumulative distribution function (CDF) of the distribution are as follows [7].

$$f(x|a, b) = ab \exp(bx - ae^{bx} + a) \quad (1)$$

$$F(x|a, b) = 1 - \exp(-ae^{bx} + a) \quad (2)$$

In equation (1) and (2), a is the shape parameter and b is the scale parameter. For scale parameter $b = 0.5$, the forms of probability density function and cumulative distribution function are shown in Figure 1 and Figure 2, respectively.

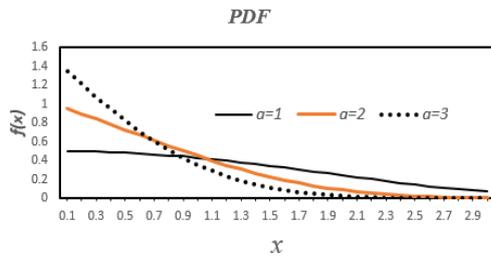


Fig. 1. Probability density function Pattern for $b = 0.5$

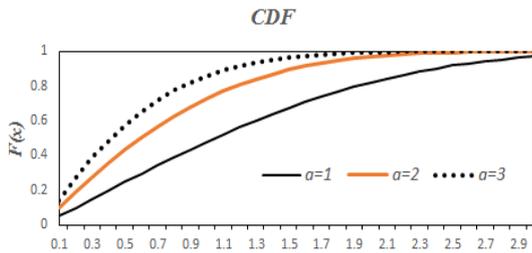


Fig. 2. Cumulative distribution function for $b = 0.5$

II.II Probability Density Function and Cumulative Distribution Function Reflecting Influential Factors

In the software testing process, learning effects can be the same or operable procedure by the testing managers, so how to use these effects is a significant request for software reliability. Influential factors, including autonomous errors-detected factor and learning factor, can be considered to detect software errors. Therefore, assuming that $f(t)$ is the probability density function that represents the probability of error found at t and $F(t)$ is the cumulative distribution function up to $(0, t]$, the model considering the influence factors is expressed as follows [8].

$$f(t) = (\gamma + \eta F(t))(1 - F(t)) \quad (3)$$

Note that $\gamma > 0, \eta > 0$.

In Equation (3), the autonomous errors-detected factor γ is not known in advance, but the automatically detects the error during the testing process and the learning factor η was sated is based on the error patterns that was founded in the past [9]. On the other hand, The Equation (3) can be converted into a hazard function form $h(t)$ as follows.

$$h(t) = (\gamma + \eta F(t)) \quad (4)$$

Note that $h(t) = \frac{f(t)}{1-F(t)}$.

Thus

$$F(t) = \frac{h(t) - \gamma}{\eta}, f(t) = F'(t) = \frac{h'(t)}{\eta} \quad (5)$$

III. NHPP SOFTWARE RELIABILITY MODEL REFLECTING EXPONENTIAL-EXPONENTIAL DISTRIBUTION INFLUENTIAL FACTORS

The hazard function of the exponential-exponential distribution is derived as follows using Equation (1) and (2) [5, 10].

$$h(t) = \frac{f(t)}{1 - F(t)} = abe^{bt} \quad (6)$$

Therefore, using Equation (5) and (6), the cumulative distribution function and probability density function [5, 10] considering the influential factors, can be derived as follows.

$$F(t) = \frac{abe^{bt} - \gamma}{\eta}, \quad f(t) = \frac{1}{\eta} ab^2 e^{bt} \quad (7)$$

Therefore, using Equation (7), the mean value function and the intensity function of the finite-failure NHPP model can be expressed as the following pattern [9].

$$m(t|\theta, a, b) = \theta F(t) = \frac{\theta}{\eta} (abe^{bt} - \gamma) \quad (8)$$

$$\lambda(t|\theta, \beta) = \theta f(t) = \frac{\theta}{\eta} ab^2 e^{bt} \quad (9)$$

Note that $t \in (0, \infty]$, a is the shape parameter and b is the scale parameter. In finite failure NHPP model, θ was specified the expected value of faults that would be discovered observing time $(0, t]$. In Equations (8) and (9), $f(t)$ is the probability density function and $F(t)$ is the cumulative distribution function.

Using Equations (8) and (9) and $\theta = \{\theta, a, b\}$ denotes the parameter space, the log-likelihood function for using the maximum likelihood estimation can be derived as follows [5, 10].

$$\begin{aligned} \ln L_{NHPP}(\theta|x) &= -m(x_n) + \left[\sum_{i=1}^n \ln \lambda(x_i) \right] \\ &= -\theta \left[\frac{abe^{bx_n} - \gamma}{\eta} \right] + \sum_{i=1}^n \ln \left[\theta \left(\frac{1}{\eta} ab^2 e^{bx_i} \right) \right] \end{aligned} \quad (10)$$

Note. $x = (x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n)$.

Therefore, while maintaining the characteristics of the exponential-exponential distribution in Equation (10), by fixing the shape parameter a for brevity, the partial differential equations for θ and b can be derived as follows [7, 8].

$$\frac{\partial \ln L_{NHPP}(\theta|x)}{\partial \theta} = \frac{n}{\theta} - \left(\frac{abe^{bx_n} - \gamma}{\eta} \right) = 0 \quad (11)$$

$$\begin{aligned} \frac{\partial \ln L_{NHPP}(\theta|x)}{\partial b} &= \frac{2n}{b} + \sum_{i=1}^n x_i \\ &\quad - \frac{\theta a}{\eta} (e^{bx_n})(1 + bx_n) = 0 \end{aligned} \quad (12)$$

Therefore, the maximum likelihood estimators $\hat{\theta}_{MLE}$ and \hat{b}_{MLE} using Equation (11) and (12) can be solved.

IV. SOFTWARE FAILURE RELIABILITY ANALYSIS REFLECTING EXPONENTIAL-EXPONENTIAL DISTRIBUTION INFLUENTIAL FACTORS

In this chapter, was used software failure time data [11] to compare and analyze the reliability characteristics of reliability models that was reflected the influencing factors. Software failure time data are summarized in Table 1 and Box-plot was used in this study for the trend test in order to detect the presence of extreme values [12, 13]. Therefore, in the result of Figure 1, three data (28th, 29th, 30th) were outliers (extreme

value) and were excluded from parameter estimation. The basic statistical results for these data are summarized in Table 2. In addition, the maximum likelihood estimation method was applied to estimate the parameters by a mathematical translation data (Failure Time $\times 10^{-2}$) to facilitate the parameter estimation. Therefore, in case of shape parameter $a = 3$, the parameter estimation results reflecting the autonomous errors-detected factor and learning effect was summarized in Table 3 when the influence factor is assumed to be 10%.

Table 1. Failure time data

Failure Number	Failure Time (hours)	Failure Time $\times 10^{-2}$	Failure Number	Failure Time (hours)	Failure Time $\times 10^{-2}$
1	30.02	0.3002	16	151.78	1.5178
2	31.46	0.3146	17	177.5	1.775
3	53.93	0.5393	18	180.29	1.8029
4	55.29	0.5529	19	182.21	1.8221
5	58.72	0.5872	20	186.34	1.8634
6	71.92	0.7192	21	256.81	2.5681
7	77.07	0.7707	22	273.88	2.7388
8	80.9	0.809	23	277.87	2.7787
9	101.9	1.019	24	453.93	4.5393
10	114.87	1.1487	25	535	5.35
11	115.34	1.1534	26	537.27	5.3727
12	121.57	1.2157	27	552.9	5.529
13	124.97	1.2497	28	673.68	6.7368
14	134.07	1.3407	29	704.49	7.0449
15	136.25	1.3625	30	738.68	7.3868

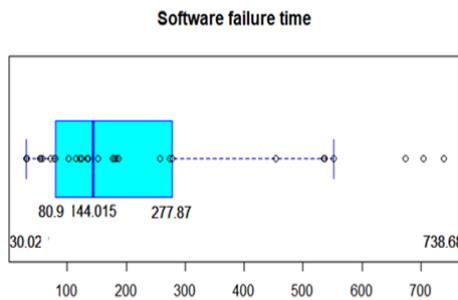


Fig. 1. Box plot test

Table 2. Basic statistics

Mean	0.1879	Skewness	1.4281
median	0.1341	range	0.5229
standard deviation	0.1568	minimum	0.03
variance	0.0246	maximum	0.5529
kurtosis	1.0371	observed Number	27

The Figure 2 also shows the mean value function pattern. Although all non-decreasing forms are shown in this Figure, it is shown that the estimated value error difference becomes smaller when the learning factor η is larger than the autonomous errors-detected factor γ in comparison with the true value.

Table 3. Parameter estimation

Model		MLE	Model Comparison	
η	γ		MSE	R^2
0.07	0.03	$\hat{\theta}_{MLE} = 1.6134$ $\hat{b}_{MLE} = 0.1628$	24.9136	0.6045
0.05	0.05	$\hat{\theta}_{MLE} = 1.2015$ $\hat{b}_{MLE} = 0.1607$	25.7719	0.5909
0.03	0.07	$\hat{\theta}_{MLE} = 0.7535$ $\hat{b}_{MLE} = 0.1586$	30.3808	0.5178

Note. η : Learning factor. γ : Autonomous errors-detected factor. MLE: Maximum likelihood estimation. MSE: Mean square error. R^2 : Coefficient of determination.

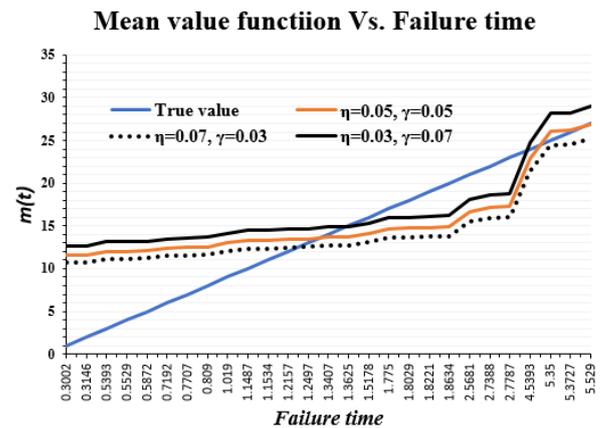


Fig. 2. Mean value function for each model

In addition, the statistics of the mean square error (MSE) [14, 15], which can be represented a measure of the difference between the actual value and the predicted value, are as follows.

$$MSE = \frac{\sum_{i=1}^n [m(x_i) - \hat{m}(x_i)]^2}{n - k} \quad (13)$$

Note that $m(x_i)$ is the cumulated true number of the faults can be perceived in $(0, x_i]$ and $\hat{m}(x_i)$ estimated number of the faults detected in $(0, x_i]$, n states the number of realizing values and k is the number of the parameter.

In Table 3, the overall mean square error is smaller when the learning factor η is larger than the autonomous errors-detected factor γ , so it can be regarded as an efficient model in terms of a measure of accuracy.

In addition, the coefficient of determination (R^2) can be defined as the explanatory tools to explain the number of the failures as the forecasting value. Thus, the model with a large coefficient of determination can be stated as an efficient model in terms of goodness-of-fit [13,14].

$$R^2 = 1 - \frac{\sum_{i=1}^n [m(x_i) - \hat{m}(x_i)]^2}{\sum_{i=1}^n (m(x_i) - \sum_{j=1}^n m(x_j)/n)^2} \quad (14)$$

Therefore, in Table 3, the estimated coefficient of determination is also larger when the learning factor η is larger

than the autonomous errors-detected factor γ . However, since all the proposed models have more than 50%, all models are judged to be efficient models in terms of goodness-of-fit [5, 12]. In the NHPP model, the software failure happens at last test failure time $x_{27} = 5.529$ and reliability which is the probability that the software failure does not occur between 5.529 and $0.529 + t$ (where t is the mission time) can be specified using the following construction [13, 14, 15].

$$\begin{aligned} \hat{R}(t|x_{27} = 5.529) &= e^{-\int_{5.529}^{5.529+t} \lambda(\eta) d\eta} \\ &= \exp[-\{m(t + 5.529) - m(5.529)\}] \end{aligned} \quad (15)$$

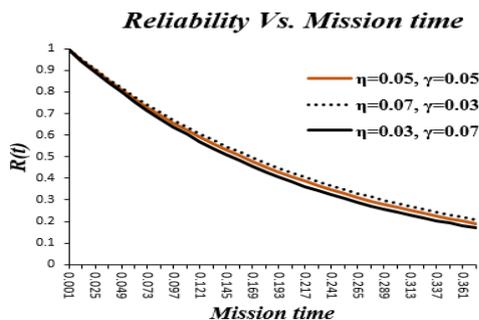


Fig. 3. Transition of reliability pattern

In the form of the reliability function in Figure 3 using the Equation (15), gradually seems as a non-increasing pattern as the mission time elapses. In case of the larger learning factor than the autonomous errors-detected factor, the higher pattern shows in terms of the reliability.

V. CONCLUSION

By quantitatively modelling the trends of failures during the software development process or during the actual software operation, the efficiency and reliability can be evaluated by comparing and analysing the accuracy and reliability of the software. The reality is that defects can hardly be avoided in the course of modifications and changes made by large software. Therefore, software managers studied NHPP software model using efficient learning process for software and test tools. The influence factors were compared and presented the problem of the characteristics of the learning effect, which is the factor set by the testing manager to detect errors in detail by prior experience and the autonomous errors-detected factor considering errors that are not known in advance. The results of this study can be summarized as follows.

First, the result of the mean value function pattern reflecting learning effect was show a non-decreasing form. If the model following greater the learning factor is than model following the autonomous errors-detected factor, the smaller estimated value error difference was happened in comparison with the true value. The overall mean square error is smaller when the learning factor is larger than autonomous errors-detected factor, so it can be regarded as an efficient model in terms of a measure of accuracy. Second, the determination coefficient estimation value is also larger when the learning factor is larger than the autonomous errors-detected factor, so the model with the large

learning factor can be considered as an efficient model in terms of goodness-of-fit. However, since all the proposed models have more than 50%, all models are judged to be efficient models. Third, in the form of the reliability function, as the mission time passes, it gradually appears as a non-increasing pattern. But, in case of the larger learning factor than autonomous errors-detected factor, the higher pattern shows in terms of the reliability. Through this study, the software operators identified the types of software failures using the learning effect feature of the exponential-exponential distribution. Thus, software developers can be used as a basic guideline to identify the learning factors depending on the life distribution and to investigate the causes of software failure.

Acknowledgements

Funding for this paper was provided by Namseoul University

REFERENCES

- [1] K. Y. Song, I. H. Chang, H. Pham, A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments. *Applied Science*. 2017; 7(983): 1-16.
- [2] S. Yamada, S. Osaki, Software reliability growth modelling: models and applications. *IEEE Transactions on Software Engineering*. 1985; 11(12): 1431-1437.
- [3] X. Teng, H. Pham, A new methodology for predicting software reliability in the random field environments. *IEEE Transactions on Reliability*. 2006; 55(3): 458-468.
- [4] Goel AL, Okumoto K., Time dependent error - detection rate model for software reliability and other performance measure. *IEEE Trans Reliability*. 1979; R-28(3): 206-211.
- [5] S. S. Gokhale, K. S. Trivedi. A time/structure based software reliability model. *Annals of Software Engineering*. 1999; 8(85): 85-121
- [6] Hee-Cheul Kim. The Comparison Analysis about Reliability Features of Software Reliability Model Using Burr-XII and Type-2 Gumbel Lifetime Distribution. *International Journal of Engineering Research and Technology*. 2019; 12(1): 73-78..
- [7] P.E. Oguntunde, O.S. Balogun, H.I. Okagbue and S.A. Bishop. The Weibull-Exponential Distribution: Its Properties and Applications. *Journal of Applied Sciences*. 2015; 15(11):1305-1311.
- [8] Kuei-Chen, C., Yeu-Shiang, H., and Tzai-Zang, L., A study of software reliability growth from the perspective of learning effects. *Reliability Engineering and System Safety* 93. 2008; 1410-1421.
- [9] Kyung-Soo Kim, Hee-Cheul Kim, The Performance Analysis of the Software Reliability NHPP Log-linear Model Depend on Viewpoint of the Learning Effects. *Indian Journal of Science and Technology*. 2017;9(37):1-7.
- [10] Hee-Cheul KIM, The Comparative Study of NHPP Half-Logistic Distribution Software Reliability Model using the Perspective of Learning Effects. *Journal of Next*

Generation Information Technology (JNIT).
2013;4(8):132-139.

- [11] Satya Prasad R, Rao KRH, Kantha RRL., Software reliability measuring using modified maximum likelihood estimation and SPC, International Journal of Computer Applications. 2011; 21(7):1-5.
- [12] Hee-Cheul Kim, A Comparative Study on Software Reliability Models with Shape Parameter of Type-2 Gumbel Life Distribution, International Journal of Soft Computing, 2017; 12(5-6):351-354.
- [13] Hee-Cheul Kim. A Comparative Study on the Finite Failure Software Reliability Model with Modified Lindley Type Lifetime Distribution. International Journal of Engineering Research and Technology. 2019; 12(6): p. 760-764.
- [14] Kim K-S, Kim H-C, The Performance Analysis of the Software Reliability NHPP Log-linear Model Depend on Viewpoint of the Learning Effects, Indian Journal of Science and Technology. 2016; 9(37):1-5.
- [15] Kim, H.C., 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, 2015; 8(34): 1-7