International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 3 (2020), pp. 438-443
© International Research Publication House. https://dx.doi.org/10.37624/IJERT/13.3.2020.438-443

A Comparative Study on the Performance Attributes of Finite Failure NHPP Software Reliability Model with Logistic Distribution Property

Tae-Jin Yang[1]

[1] *Professor, Department of Electronic Engineering, Namseoul University, South Korea.*

ORCID:0000-0001-7971-277X

Abstract

In this study, the performance of reliability is analyzed by applying the Logistic distribution property to the finite-fault NHPP reliability model. For this, software failure time data was used, parametric estimation was applied to the maximum likelihood estimation method (MLE), and nonlinear equations were calculated using the bisection method. As a result, in the analysis of the intensity function, the Log-Logistic model is efficient because the failure occurring rate decreases with the failure time and the mean square error (MSE) is small. In the analysis of the mean value function, all the proposed models showed a slightly underestimated value compared to the true value, but the Log-Logistic model showed the smallest error value to the true value along with the Goel-Okumoto model. As a result of evaluating the software reliability after putting the mission time in the future, the Log-Logistic model shows a higher reliability trend than the other models in which the reliability decreases with the mission time. In conclusion, we found that the Log-Logistic model is an efficient model with the best performance among the proposed models. In this study, the reliability performance of the Logistic type distribution model without the existing research case was newly analyzed, and it is expected that it can be used as a basic guideline for the software developers to search the optimal software reliability model.

Keywords: NHPP Model, Logistic Distribution Property, Log-Logistic Distribution, Half-Logistic Distribution, Software Reliability, Reliability Performance Analysis.

I. INTRODUCTION

Software technology, which is the core of the 4th industrial revolution era, has spread rapidly in various industrial fields, and the need for software quality that can process a large amount of information without failures is also increasing. To solve this problem, software developers are still doing a lot of research to search for ways to improve software reliability. For this reason, software reliability models using the non-homogeneous Poisson process (NHPP) have been extensively studied to improve software reliability. In particular, many NHPP software reliability models using the intensity function and the mean value function have been proposed to estimate the reliability attributes such as the number of residual failures and the failure rate in a controlled test environment [1]. Concerning the NHPP reliability model, Goel and Okumoto [2] proposed an exponential type software reliability model, Huang [3] explained the software reliability attributes using the mean value function, Shyur [4] proposed a generalized reliability model using change-point. Also, Kim [5] analyzed the attributes of software reliability based on the finite failure software reliability model with modified Lindley type lifetime distribution. Also, Pham and Zhang [6] proposed a new model based on NHPP software reliability with testing coverage, and Voda [7] proposed that various types of lifetime distributions can be explained by the inverse-Rayleigh distribution. Yang [8] also proposed a new performance analysis results of finite failure NHPP software reliability model based on Weibull lifetime distribution.

Therefore, in this study, after applying the Logistic distribution widely used in the reliability field to the finite-fault NHPP model, we were newly analyzed the reliability performance of the proposed models and will present the optimal software reliability model through this analysis results.

II. RELATED RESEARCH

II. I NHPP Software Reliability Model

N(t) is the cumulative number of failures of the software detected up to time t, m(t) is a mean value function when λ(t) is expressed by an intensity function, the cumulative failure number N(t) follows a Poisson probability density function having a parameter m(t). The software reliability model of the non-homogeneous Poisson process (NHPP) is a model that measures the reliability by using the average failure rate function around the number of failures generated per unit time.

That is

$$\mathrm{P}\{N(t)=n\}=\frac{[m(t)]^{n}\cdot e^{-m(t)}}{n!} \tag{1}$$

Note that $n = 0,1,2,\cdots\infty$.

The mean value function m(t) and the intensity function λ(t) of the NHPP model are as follows.

$$\mathrm{m(t)} = \int_0^t \lambda(s)ds \tag{2}$$

$$\frac{dm(t)}{d(t)} = \lambda(t) \tag{3}$$

In terms of software reliability, the mean value function represents a software failure occurrence expected value, the intensity function is the failure rate function and means the failure occurrence rate per defect. Also, the time domain NHPP models are classified into a finite failure that the failure does not occur at the time of repairing the failure, and an infinite failure that the failure occurs at the time of repairing failure. In this study, we will analyze the software reliability performance based on finite failure cases. That is, in the finite-failure NHPP model, if the expected value of the failure that can be found up to time [0, t] is θ, then the mean value function and the intensity function are as follows.

$$\mathrm{m(t}|\theta,\mathrm{b)} = \theta F\mathrm{(t)} \tag{4}$$

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

Considering the failure time data up to the n^{th} and the Eq. 4 and Eq. 5, the likelihood function of the finite-failure NHPP model is derived as follows.

$$L_{NHPP}(\Theta|\underline{x}) = \left(\prod_{i=1}^{n} \lambda(x_i)\right) exp[-m(x_n)] \tag{6}$$

Note that $\underline{x} = (x_1, x_2, x_3 \cdots x_n)$.

II.II Finite Failure NHPP : Goel-Okumoto Basic Model

The Goel-Okumoto model is a well-known basic model in the software reliability field. Let f(t) and F(t) for the Goel-Okumoto model be a probability density function and a cumulative density function, respectively. Assuming that the expected value of the number of failures of the observation point [0, t] is θ, the finite failure strength function and the mean value function are as follows.

$$\mathrm{m(t}|\theta,\mathrm{b)} = \theta F\mathrm{(t)} = \theta(1 - e^{-bt}) \tag{7}$$

$$\lambda(\mathrm{t}|\theta,\mathrm{b}) = \theta \mathrm{f(t)} = \theta \mathrm{b} e^{-bt} \tag{8}$$

Note that $\theta > 0,\ \mathrm{b} > 0$.

Considering the failure time data up to the n^{th} and the Eq. 7 and Eq. 8, the likelihood function of the finite-failure NHPP model is derived as follows.

$$L_{NHPP}(\theta, \mathrm{b}|\underline{x}) = \left(\prod_{i=1}^{n} \theta b e^{-bx_i}\right) exp[-\theta(1 - e^{-bx_n})] \tag{9}$$

Note that $\underline{x} = (0 \le x_1 \le x_2 \le \cdots \le x_n)$.

The log-likelihood function, using the Eq. 9, is simplified to the following log conditional expression.

$$\ln L_{NHPP}(\Theta|\underline{x}) = nln\theta + nlnb - b\sum_{k=1}^{n} x_k - \theta(1 - e^{-bx_n}) \tag{10}$$

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ satisfying the following Eq. 11 and Eq. 12 can be estimated by a numerical method.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - 1 + e^{-\hat{b}x_n} = 0 \tag{11}$$

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial b} = \frac{n}{\hat{b}} - \sum_{i=1}^{n} x_n - \hat{\theta} x_n e^{-\hat{b}x_n} = 0 \tag{12}$$

II.III. Finite Failure NHPP : Half-Logistic Model

The Half-Logistic distribution is widely known as a suitable model for process controls and reliability measurements.

The probability density function and the cumulative distribution function of this distribution are as follows [9].

$$\mathrm{f(t}|\theta,\beta) = \frac{2\beta e^{-\beta t}}{(1 + e^{-\beta t})^2} \tag{13}$$

$$F(\mathrm{t}|\theta,\beta) = \frac{(1 - e^{-\beta t})}{(1 + e^{-\beta t})} \tag{14}$$

Note that $\beta > 0,\ \mathrm{t} \in [0,\ \infty]$.

Therefore, the mean value function and the intensity function of the finite fault NHPP Half-Logistic model are as follows.

$$\mathrm{m(t}|\theta,\ \beta) = \theta F\mathrm{(t)} = \theta \frac{(1 - e^{-\beta t})}{(1 + e^{-\beta t})} \tag{15}$$

$$\lambda(\mathrm{t}|\theta,\beta) = \theta \mathrm{f(t)} = \theta \frac{2\beta e^{-\beta t}}{(1 + e^{-\beta t})^2} \tag{16}$$

Note that $\theta > 0,\ \beta > 0$.

Therefore, after replacing t with the final point of failure x_n, the likelihood function to maximum likelihood estimation (MLE) by using Eq. 15 and Eq. 16 is derived as follows.

$$L_{NHPP}(\Theta|D_{x_n}) = \left(\prod_{i=1}^{n} \theta \frac{2\beta e^{-\beta x_i}}{(1 + e^{-\beta x_i})^2}\right) \exp\left[-\theta \frac{(1 - e^{-\beta x_n})}{(1 + e^{-\beta x_n})}\right] \tag{17}$$

Note that Θ is parameter space.

Therefore, when Eq. 17 is partially differentiated from parameters θ and β, the maximum likelihood estimator $\hat{\theta}_{MLE}$

and $\hat{\beta}_{MLE}$ satisfy the following the Eq. 18.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial\theta} = 0, \qquad \frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial\beta} = 0$$

$$\frac{\partial^2 \ln L_{NHPP}(\Theta|\underline{x})}{\partial^2\theta} = 0 \tag{18}$$

Note that $\underline{x} = (x_1, x_2, x_3 \cdots x_n)$.

Therefore, substituting Eq. 17 into Eq. 18 is as follows.

$$\hat{\theta} = \frac{\left(1 + e^{-\beta x_n}\right)}{(1 - e^{-\beta x_n})} \tag{19}$$

$$g(\beta) = \sum_{i=1}^{n} x_i - \frac{n}{\beta} - 2\sum_{i=1}^{n-1}\left[\frac{x_i\, e^{-\beta x_i}}{(1 + e^{-\beta x_i})}\right]$$

$$- \frac{2\, x_n\, e^{-\beta x_n}}{(1 + e^{-\beta x_n})}\left[1 - \left(\frac{n}{1 - e^{-\beta x_n}}\right)\right] = 0 \tag{20}$$

Note that $x = \left(x_{1,}\, x_{2,},\ x_3 \cdots\ x_n\right)$.

Therefore, in Eq. 20, the maximum likelihood estimator $\hat{\beta}_{MLE}$ is calculated using the Newton-Raphson method, and then the maximum likelihood estimator $\hat{\theta}_{MLE}$ can be calculated by substituting $\hat{\beta}_{MLE}$ into Eq. 19.

II.IV. Finite Failure NHPP : Log-Logistic Model

The Log-Logistic distribution has a property that increases and decreases in the form of failure rate and thus is widely applied in the reliability field. The probability density function and the cumulative distribution function considering the shape parameter(k) are as follows [10].

$$\mathrm{f(t}|\tau, \mathrm{k)} = \frac{\tau k(\tau t)^{k-1}}{[1 + (\tau t)^k]^2} \tag{21}$$

$$\mathrm{F(t}|\tau, \mathrm{k)} = \frac{(\tau t)^k}{[1 + (\tau t)^k]} \tag{22}$$

Note that $\tau > 0,\ \mathrm{k} > 0$

Therefore, the mean value function and the intensity function of the finite fault NHPP Log-Logistic model are as follows.

$$\mathrm{m(t}|\theta, \tau, \mathrm{k)} = \theta F(\mathrm{t}) = \theta\ \frac{(\tau t)^k}{[1 + (\tau t)^k]} \tag{23}$$

$$\lambda(\mathrm{t}|\theta, \tau, \mathrm{k}) = \theta \mathrm{f(t)} = \theta\ \frac{\tau k(\tau t)^{k-1}}{[1 + (\tau t)^k]^2} \tag{24}$$

Note that $\theta > 0,\ \tau, \mathrm{k} > 0$.

Since the Log-Logistic distribution has a form in which the failure rate increases, the case where the shape parameter that determines the shape of the failure lifetime distribution is 2 was applied. Therefore, after replacing t with the final point of failure x_n, the log-likelihood function to maximum likelihood estimation (MLE) by using Eq. 23 and Eq. 24 is derived as follows.

$$\ln L_{NHPP}(\Theta|\underline{x}) = nln2 + nln\theta + 2nln\tau$$

$$+\sum_{i=1}^{n} x_i - 2\sum_{i=1}^{n} ln[1 + (\tau x_i)^2] - \theta\,\frac{(\tau x_n)^2}{[1 + (\tau x_n)^2]} = 0 \tag{25}$$

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and $\hat{\tau}_{MLE}$ satisfying the following Eq. 26 and Eq. 27 can be estimated by a numerical method.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial\theta} = \frac{n}{\hat{\theta}} - \frac{(\hat{\tau} x_n)^2}{[1 + (\hat{\tau} x_n)^2]} = 0 \tag{26}$$

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial\tau} = \frac{2n}{\hat{\tau}} - 2\hat{\tau}\sum_{i=1}^{n} {x_i}^2\ \frac{1}{ln[1 + (\hat{\tau} x_i)^2]}$$

$$-\hat{\theta}\ \left(\frac{2\hat{\tau}{x_n}^2(1 + \hat{\tau}^2\, x_n - \hat{\tau}^2\, {x_n}^2}{[1 + (\hat{\tau} x_n)^2]^2}\right) = 0 \tag{27}$$

Note that $x = \left(x_{1,}\, x_{2,},\ x_3 \cdots\ x_n\right)$.

III. THE PROPOSED ANALYSIS ALGORITHM AND SOLUTIONS

The analysis algorithm of the proposed software reliability model is as follows.

Step 1: Validating the software failure data collected through the Laplace trend test analysis.

Step 2: Calculating the parameters$\left(\hat{\theta}, \hat{\beta}, \hat{\tau}\right)$ for the proposed model using the maximum likelihood estimation (MLE).

Step 3: Calculating the coefficient of determination (R^2) and mean square error (MSE) for efficient model selection.

Step 4: Analyzing the performance attributes ($m(t)$, $\lambda(t)$) and reliability$\left(\hat{R}(\tau)\right)$ of proposed models.

Step 5: Providing research information on the optimal model by analyzing the performance of the proposed model.

After analyzing the performance of the proposed model using the above steps, we will present information on the optimal model that software developers need.

Let compare and analyze the performance of the proposed reliability models using the software failure time data [11] as shown in Table 1. This software failure time is the data that was occurred 30 times in 738.68 unit time.

Table 1. Software Failure Time Data

Failure Number	Failure Time (hours)	Failure Number	Failure Time (hours)
1	30.02	16	151.78
2	31.46	17	177.50
3	53.93	18	180.29
4	55.29	19	182.21
5	58.72	20	186.34
6	71.92	21	256.81
7	77.07	22	273.88
8	80.90	23	277.87
9	101.90	24	453.93
10	114.87	25	535
11	115.34	26	537.27
12	121.57	27	552.90
13	124.97	28	673.68
14	134.07	29	704.49
15	136.25	30	738.68

Laplace trend test was used to verify the reliability of the software failure time data as shown in Figure 1.

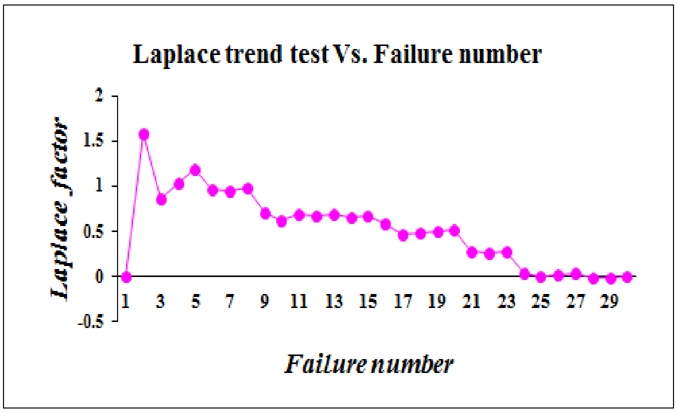

Fig. 1. Estimation Results of Laplace Trend Test

In general, if the Laplace factor estimates are distributed between -2 and 2, the data are reliable because the extreme values do not exist and are stable.

As a result of this test, the estimated value of the Laplace factor was distributed between 0 and 2, as shown in Figure 1. Therefore, it is possible to apply this data because there is no extreme value [12].

In this study, the maximum likelihood estimation (MLE) was used to perform parameter estimation. The calculation method of the nonlinear equations is solved using the bisection method, and the results are shown in Table 2.

Table 2. Parameter Estimation of Each Model

Model	MLE		Model comparison	
			MSE	R^2
Goel-Okumoto	$\hat{\theta} = 31.8596$	$\hat{\beta} = 0.00309$	7.66779	0.976
Half-Logistic	$\hat{\theta} = 31.2769$	$\hat{\beta} = 0.00433$	16.39770	0.948
Log-Logistic	$\hat{\theta} = 32.2412$	$\hat{\tau} = 0.00495$	8.97306	0.972

Explanatory notes.

MLE = Maximum Likelihood Estimation
MSE = Mean Square Error,
$R^2 = Coefficient\ of\ Determination.$

As the basis for determining the efficient model, the mean square error is defined as follows.

$$MSE = \frac{\sum_{i=1}^{n} \left(m(x_i) - \hat{m}(x_i)\right)^2}{n-k} \quad (28)$$

Note that $m(x_i)$ is the total accumulated number of errors observed within time is $(0, x_i)$, $\hat{m}(x_i)$ is the estimated cumulative number of errors at time x_i obtained from the fitting mean value function, n is the number of observations and k is the number of parameters to be estimated. When selecting an efficient model, the smaller the mean square error, the more efficient the model.

The coefficient of determination (R^2) is a measuring value to the explanatory power of the difference between the target value and the observed value. When selecting an efficient model, the larger the value of the decision coefficient, the more efficient the model because the error is relatively small. It is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(m(x_i) - \hat{m}(x_i)\right)^2}{\sum_{i=1}^{n} \left(m(x_i) - \sum_{j=1}^{n} m(x_j)/n\right)^2} \quad (29)$$

As shown in Table 2, we can see that the Log-Logistic model is more efficient than the Half-Logistic model. But, the Goel-Okumoto model has the largest coefficient of determination and the smallest mean square error is more efficient than the other models [13].

Figure 2 shows the transition of mean square error (MSE) according to each failure number. Also, in this figure, the Log-Logistic model shows better estimates than the other models in the total range of failure numbers.

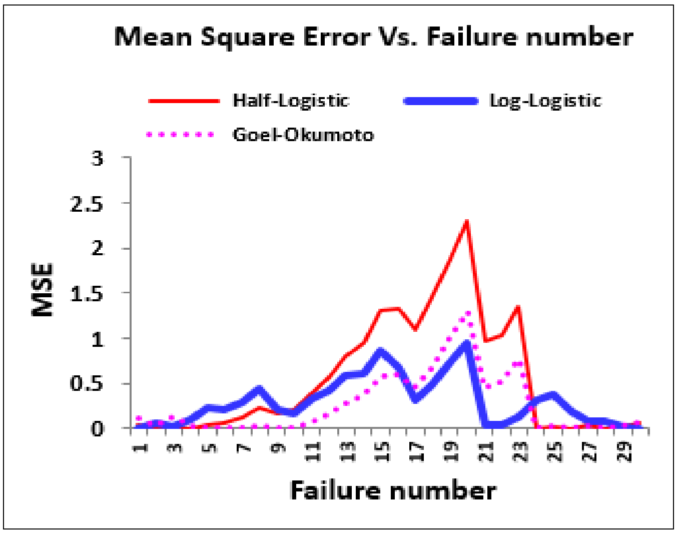

Fig. 2. Transition of Mean Square Error

In Figure 2, the mean squared error of the Goel-Okumoto model shows a trend of the smallest error with time, which is more efficient than the other models in terms of fitness.

Figure 3 shows trends in the strength function, which is the failure occurring rate per defect. The Log-Logistic model shows the greatest decreasing tendency as the failure time passes, indicating that it is an efficient model, and the Half-Logistic model also shows a similar pattern.

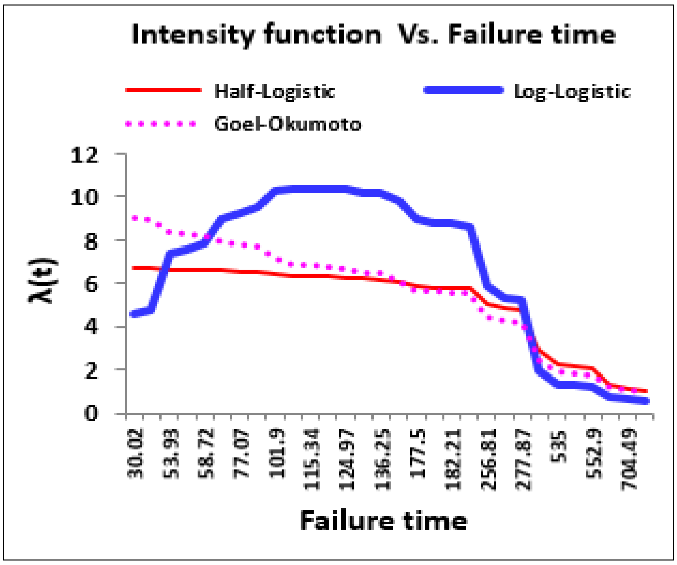

Fig. 3. Transition of Intensity Function λ(t)

Figure 4 shows the pattern trend for the mean value function, which is the failure occurring expected value. In this figure, all models show underestimated from the difference between the true values, and the Goel-Okumoto model has the smallest underestimated pattern.

Also, the Log-Logistic model is more efficient than the Half-Logistic model because the error width is small.

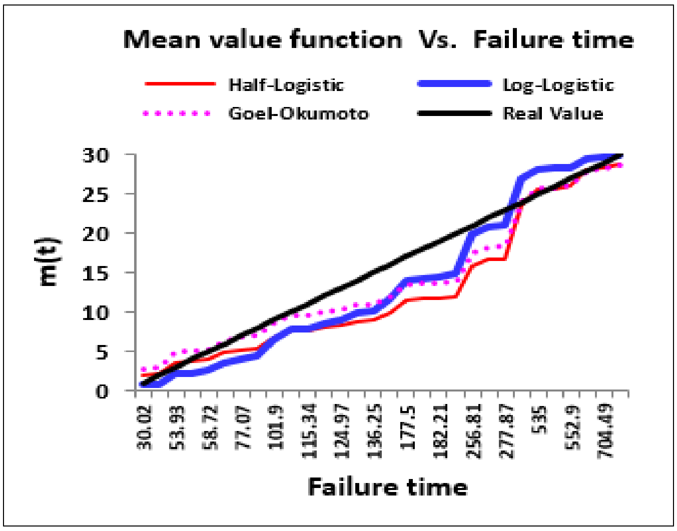

Fig. 4. Pattern of Mean Value Function $m(t)$

Let analyze the reliability performance of the proposed models for future mission time. Here, reliability is the probability that a software failure will occur when testing at $x_n = 738.68$, and no software failure will occur between confidence intervals $[\ x_n,\ x_n + \tau]$ where τ is the future mission time. Therefore, the reliability of future mission time is as follows [14].

$$\hat{R}(\tau|x_n) = e^{-\int_{x_n}^{x_n+\tau}\lambda(\tau)d\tau}$$

$$= \exp[-\{m(\tau + x_n) - m(x_n)\}]$$

$$= \exp[-\{m(\tau + 738.68) - m(738.68)\}] \qquad (30)$$

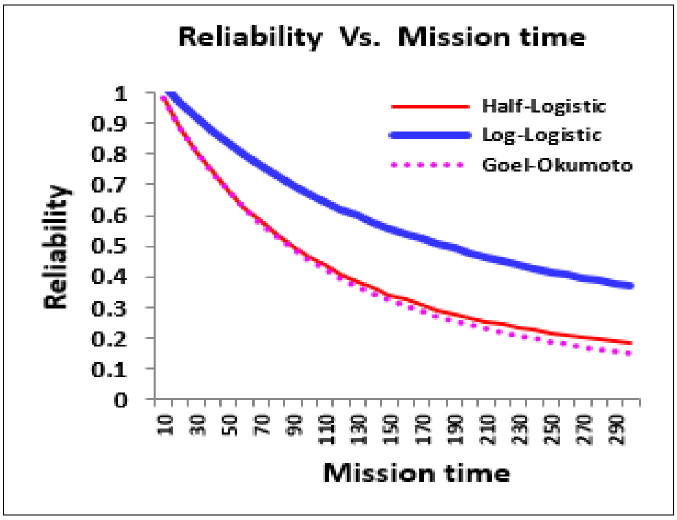

Fig. 5. Transition of Reliability $\hat{R}(t)$

As shown in Figure 5, the Log-Logistic model shows a higher reliability trend than the other models in which the reliability decreases with the mission time. That is, the Log-Logistic model is more efficient than the other models because the reliability is the highest.

IV. CONCLUSION

It is possible to efficiently improve the reliability performance by analyzing the performance after quantitatively modeling the occurrence of the failure in the software test operation or the software development process. In this study, based on the finite-fault NHPP model with software failure time data, we compared and analyzed the software reliability performance of the Log-Logistic model and the Half-Logistic model with Logistic distribution property, together with Goel-Okumoto basic model.

The results of this study can be summarized as follows.

First, In the performance analysis of the strength function, the Log-Logistic model shows the greatest decreasing tendency as the failure time passes, indicating that it is an efficient model, and the mean square error (MSE) showed the smallest trend along with the Goel-Okumoto model.

Second, in the performance analysis of the mean value function, all the proposed models showed underestimation patterns in the error estimation for true values, and the Goel-Okumoto model has the smallest error estimation. Also, the Log-Logistic model is more efficient than the Half-Logistic model because of the small error width.

Third, in the performance analysis of mission reliability, the Log-Logistic model shows a higher reliability trend than the other models in which the reliability decreases with the mission time. That is, the Log-Logistic model has the best performance than other models because the reliability is the highest.

Fourth, a comprehensive analysis of these simulations results shows that the Log-Logistic model is an efficient model with the best performance among the proposed models.

As a result, through this study, along with a new analysis on the reliability performance of the proposed model without existing research examples, we were able to provide the research information that software developers can use as a basic design guideline. Also, future research will be needed to find the optimal model through the reliability performance analysis after applying the same type of software failure time data to various reliability models.

ACKNOWLEDGEMENTS

Funding for this paper was provided by Namseoul University.

REFERENCES

[1] Gokhale, S. S. and Trivedi, K. S. A, Time/structure based software reliability model. Annals of Software Engineering 8. 1999;5-12.

[2] Goel, A. L, Okumoto K, Time-dependent fault detection rate model for software and other performance measures. IEEE Trans. Reliability. 1978;28:206-211.

[3] C. Y. Huang, Performance analysis of software reliability growth models with testing-effort and change-point. Journal of Systems and Software. 2005;76(2):181-194.

[4] Shyur H-J, A stochastic software reliability model with imperfect debugging and change-point. J. Syst. Software 66. 2003;135-141.

[5] 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): 760-764.

[6] Pham H, Zhang X., NHPP Software Realiability and Cost Models with testing coverage. Eur.J. Oper. Res. 2003; 145:445-454.

[7] Voda, R. GH., On the inverse Rayleigh variable. Rep.Stat. Apph. Res. Juse, 1972;19(4):15-21.

[8] Tae-Jin Yang, A Study on the Reliability Performance Analysis of Finite Failure NHPP Software Reliability Model Based on Weibull Life Distribution. International Journal of Engineering Research and Technology. 2019; 12(11):1890-1896.

[9] Tae-Jin. Yang, Jea-Gun Park, A comparative study of the Software NHPP based on weibull extension distribution and flexible weibull extension distribution. International Journal of Soft Computing. 2016;11(4):276-281.

[10] Adhikari, T. R. and Srivastava, R.S., Poisson-Size biased Lindley Distribution. International Journal of Scientific and Research Publications. 2014;4(1):1-6.

[11] Y. Hayakawa, G. Telfar, Mixed Poisson-type processes with application in software reliability. Mathematical and Computer Modelling. 2000;31:151-156.

[12] Tae-Jin Yang, A Comparative Study on Reliability Attributes of Software Reliability Model Based on Type 2 Gumbel and Erlang Life Distribution. Journal of Engineering and Applied Sciences. 2019;14(10):3366-3370.

[13] Shanker, R., Shanker distribution and Its Applications. International Journal of Statistics and Applications. 2015;5(6):338-348.

[14] Tae-Jin Yang, The Analysis and Predict of Software Failure Time Based on Nonlinear Regression. Journal of Engineering and Applied Sciences. 2018;13(12):4376 -4380.