International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 11 (2019), pp. 1890-1896

A Study on the Reliability Performance Analysis of Finite Failure NHPP Software Reliability Model Based on Weibull Life Distribution

Tae-Jin, Yang
Namseoul University, Department of Electronic Engineering, Chungnam, 31020, Korea.
**Corresponding author: solomon645@nsu.ac.kr*

Abstract:

In this study, the performance of reliability is analyzed by applying the Weibull life distribution 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, and nonlinear equations were calculated using the bisection method. As a result, in the analysis of the intensity function, the Inverse-exponential model is an efficient model because the failure occurring rate decreases with the failure time and the mean square error (MSE) is the smallest. In the analysis of the mean value function, all the proposed models showed a slightly overestimated value compared to the true value, but the Inverse-exponential model showed the smallest error value to the true value. As a result of evaluating the software reliability after putting the mission time in the future, the Inverse-exponential model was stable and high together with the Rayleigh model, but the Goel-Okumoto basic model showed a decreasing tendency. In conclusion, we found that the Inverse-exponential model is an efficient model with the best performance among the proposed models. In this study, the reliability performance of the Weibull life 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 exploring the optimal software reliability model.

Keyword: Finite Failure NHPP, Inverse-exponential Distribution, Rayleigh Distribution, Software Reliability Performance Analysis, Weibull Life Distribution.

1. 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 development that can process large amount of information quickly and accurately without failures is also increasing. To solve this problem, software developers are still doing a lot of research to search ways to improve software reliability.

After all, for software developers, solving problems to improve software reliability is a very important topic. 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 failure and the failure rate in a controlled test environment [1]. In relation to the NHPP reliability model, Goel and Okumoto [2] proposed an exponential 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, and Kim [5] analyzed the attributes of software development costs based on the weibull distribution. In addition, 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, Zhang and Wu [8] also proposed a new software reliability cost model based on software failure time.

Therefore, in this study, after applying the Weibull distribution widely used in the reliability field to the finite-fault NHPP model, we analyze the reliability performance of the NHPP Weibull reliability model, and will present the optimal software reliability model through the analysis results.

2. RELATED RESEARCH

2.1 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 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 $\mathrm{m(t)}$ 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.

$$m(t|\theta, b) = \theta F(t) \quad (4)$$

$$\lambda(t|\theta, b) = \theta F(t)' = \theta f(t) \quad (5)$$

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 :

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

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

2.2 Finite Failure NHPP : Goel-Okumoto Basic Model

The Goel-Okumoto model is a well-known basic model in 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.

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

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

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

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 :

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

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

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

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

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

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

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

2.3 Finite Failure NHPP : Rayleigh Model

The Weibull lifetime distribution is widely known as a suitable model for life test and reliability measurements.

The probability density function and the cumulative distribution function considering the shape parameter(α) are as follows [9].

$$f(t) = \frac{t^{\alpha-1}}{\beta^2} e^{-\frac{t^\alpha}{2\beta^2}} \quad (13)$$

$$F(t) = \left(1 - e^{-\frac{t^\alpha}{2\beta^2}}\right) \quad (14)$$

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

In order to simplify the Eq. 13 and Eq. 14, if substitution by the equation $\frac{1}{2\beta^2} = b$ is as follows.

$$f(t) = 2bt^{\alpha-1}e^{-bt^\alpha} \quad (15)$$

$$F(t) = \left(1 - e^{-bt^\alpha}\right) \quad (16)$$

Note that $b > 0,\ t \in [0, \infty]$.

In the Weibull distribution such as Eq. 15 and Eq. 16, an exponential distribution is obtained when the shape parameter(α) is 1, and a Rayleigh distribution is obtained when the shape parameter(α) is 2. Therefore, the mean value function and the intensity function of the finite fault NHPP Rayleigh model are as follows.

$$m(t|\theta, b) = \theta F(t) = \theta(1 - e^{-bt^2}) \quad (17)$$

$$\lambda(t|\theta, b) = \theta f(t) = 2\theta b t e^{-bt^2} \quad (18)$$

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

The log-likelihood function to Maximum Likelihood Estimation (MLE) by using the Eq. 17 and Eq. 18 is derived as follows.

$$\ln L_{NHPP}(\Theta|\underline{x}) = nln2 + nln\theta + nlnb + \sum_{i=1}^{n} lnx_i \qquad (19)$$

$$-b\sum_{i=1}^{n} {x_i}^2 - \theta\left(1 - e^{-bx_n^2}\right)$$

Note that $\underline{x} = 0 \le x_1 \le x_2 \le \cdots \le x_n$, Θ is parameter space.

The partial derivatives of the parameters θ and b are as follows.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - 1 + e^{-\hat{b}x_n^2} = 0 \qquad (20)$$

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial b} = \frac{n}{\hat{b}} - \sum_{i=1}^{n} x_i^2 - \hat{\theta}{x_n}^2 e^{-\hat{b}x_n^2} = 0 \qquad (21)$$

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

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ satisfying the following the Eq. 20 and Eq. 21 can be estimated by a numerical method.

2.4 Finite Failure NHPP : Inverse-Exponential Model

The Inverse-weibull distribution is a widely applied distribution in reliability and medical field. In this Inverse-weibull distribution, the Inverse-exponential distribution is obtained when the shape parameter (γ) is 1. Here, the cumulative distribution function F(t) of the Inverse-weibull distribution is as follows[10].

$$F(\mathrm{t}) = e^{-(bt)^{-\gamma}} \qquad (22)$$

Note that $b > 0$, $t \in [0, \infty]$, γ is shape parameter.

Therefore, when the shape parameter conditions ($\gamma = 1$) are applied in the Inverse-weibull distribution, the probability density function f(t) and the cumulative distribution function F(t) of the Inverse-exponential distribution are as follows.

$$\mathrm{f(t)} = \mathrm{F(t)}' = b^{-1}t^{-2}e^{-(bt)^{-1}} \qquad (23)$$

$$\mathrm{F(t)} = e^{-(bt)^{-1}} \qquad (24)$$

Note that $b > 0$, $t \in [0,\ \infty]$.

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

$$\mathrm{m}(\mathrm{t}|\theta, \mathrm{b}) = \theta F(\mathrm{t}) = \theta e^{-(bt)^{-1}} \qquad (25)$$

$$\lambda(\mathrm{t}|\theta, \mathrm{b}) = \theta \mathrm{f(t)} = \theta b^{-1}t^{-2}e^{-(bt)^{-1}} \qquad (26)$$

The log-likelihood function to Maximum Likelihood Estimation (MLE) by using the Eq. 25 and Eq. 26 is derived as follows.

$$\ln L_{NHPP}(\Theta|\underline{x}) = nln\theta - nlnb \qquad (27)$$

$$+2\sum_{i=1}^{n} x_i - \sum_{i=1}^{n}(bx_i)^{-1} - \hat{\theta}e^{-(bx_n)^{-1}} = 0$$

The partial derivatives of the parameters θ and b are as follows.

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - e^{-(\hat{b}x_n)^{-1}} = 0 \qquad (28)$$

$$\frac{\partial \ln L_{NHPP}(\Theta|\underline{x})}{\partial b} = -\frac{n}{\hat{b}} + \frac{1}{\widehat{b^2}}\sum_{i=1}^{n}\frac{1}{x_i} \qquad (29)$$

$$-\theta\frac{1}{b^2 x_n}e^{-(\hat{b}x_n)^{-1}} = 0$$

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

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ satisfying the following the Eq. 28 and Eq. 29 can be estimated by a numerical method.

3. 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 $(\hat{\theta}, \hat{b})$ for the proposed model using the Maximum Likelihood Estimation.

Step 3: Calculating coefficient of determination (R^2) and Mean Square Error(MSE) for efficient model selection.

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

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

After analyzing the performance of the proposed model using the above steps, we will present information on the 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 is the data that was occurred 30 times in 187.35 unit time.

International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 11 (2019), pp. 1890-1896

Table 1. Software Failure Time Data

Failure number	Failure time (hours)	Failure interval-time	Failure time (hours)× 10^{-1}
1	4.79	4.79	0.479
2	7.45	2.66	0.745
3	10.22	2.77	1.022
4	15.76	5.54	1.576
5	26.10	10.34	2.610
6	35.59	9.49	3.559
7	42.52	6.93	4.252
8	48.49	5.97	4.849
9	49.66	1.17	4.966
10	51.36	1.70	5.136
11	52.53	1.17	5.253
12	65.27	12.74	6.527
13	69.96	4.69	6.996
14	81.70	11.74	8.170
15	88.63	6.93	8.863
16	107.71	19.08	10.771
17	109.06	1.35	10.906
18	111.83	2.77	11.183
19	117.79	5.96	11.779
20	125.36	7.57	12.536
21	129.73	4.37	12.973
22	152.03	22.30	15.203
23	156.40	4.37	15.640
24	159.80	3.40	15.980
25	163.85	4.05	16.385
26	169.60	5.75	16.960
27	172.37	2.77	17.237
28	176.00	3.63	17.600
29	181.22	5.22	18.122
30	187.35	6.13	18.735

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

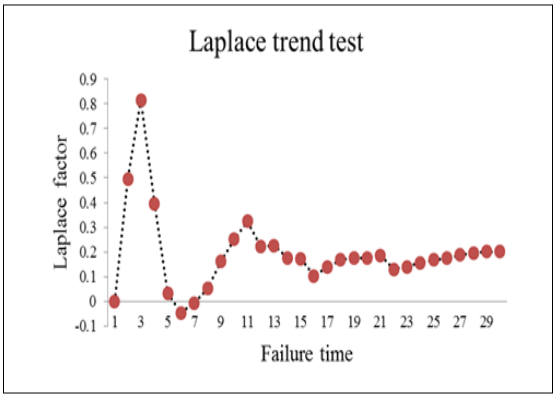

Fig 1. Estimation Results of Laplace Trend Test

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 in this Fig 1, the estimated value of the Laplace factor was distributed between 0 and 2, as shown in Fig. 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. And numerical conversion data (Failure time[hours] $\times 10^{-1}$) in order to facilitate the parameter estimation was used. 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	$\hat{\theta}$ =32.9261	$\hat{b}$ =0.1297	32.9379	0.8956
Rayleigh	$\hat{\theta}$ =30.0412	$\hat{b}$ =0.0188	32.1798	0.8980
Inverse-exponential	$\hat{\theta}$ =41.2881	$\hat{b}$ =0.1692	20.2035	0.9359

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(MSE) is defined as follows :

$$MSE = \frac{\sum_{i=1}^{n} \left(\mathrm{m}(x_i) - \widehat{\mathrm{m}}(x_i)\right)^2}{n-k} \tag{30}$$

Note that $\mathrm{m}(x_i)$ is the total accumulated number of errors observed within time is $(0, x_i)$, $\widehat{\mathrm{m}}(x_i)$ 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. In efficient model selection, 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. The larger the value of the decision coefficient in efficient model selection, the more efficient the model because the error is relatively small. It is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(\mathrm{m}(x_i) - \widehat{\mathrm{m}}(x_i)\right)^2}{\sum_{i=1}^{n} \left(\mathrm{m}(x_i) - \sum_{j=1}^{n} m(x_j)/n\right)^2} \tag{31}$$

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

Also, Fig 2 shows the transition of mean square error (MSE) according to each failure number. That is, in this figure, the Inverse-exponential model shows better estimates than the other model in the total range of failure number.

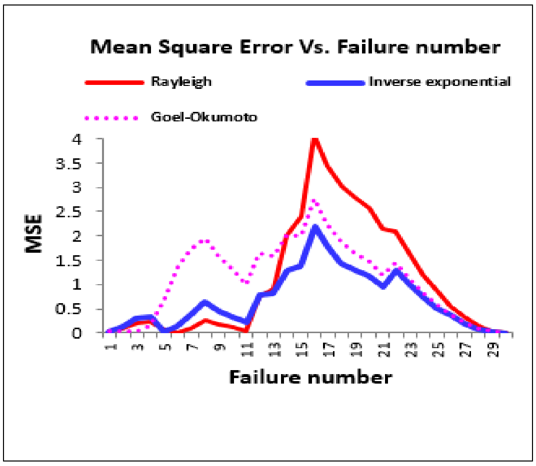

Fig 2. Transition of Mean Square Error

In this Fig 2, the mean squared error of the Inverse-exponential model shows a trend of the smallest error with time, which is more efficient than the other model in terms of fitness.

Fig 3 shows trends in the strength function, which is the failure occurring rate per defect. The Rayleigh model shows the greatest decreasing tendency as the failure time passes, indicating that it is an efficient model, but the Inverse-exponential model also shows a similar pattern, indicating that it is efficient.

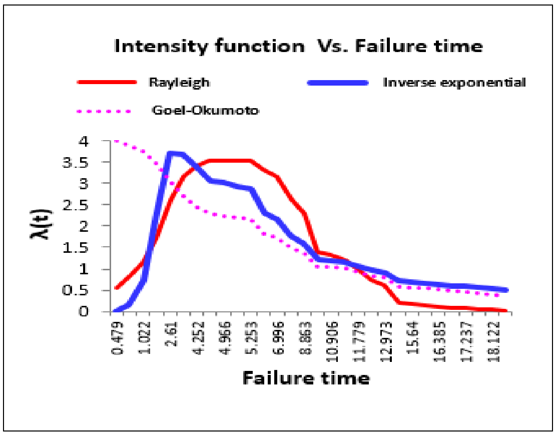

Fig 3. Transition of Intensity Function λ(t)

Fig 4 shows the pattern trend for the mean value function, which is the failure occurring expected value. In this figure, the all models shows overestimated from the difference between the true values, but the Inverse-exponential model has the smallest overestimated pattern. That is, the Inverse-exponential model is more efficient than the other model because the error width is the smallest.

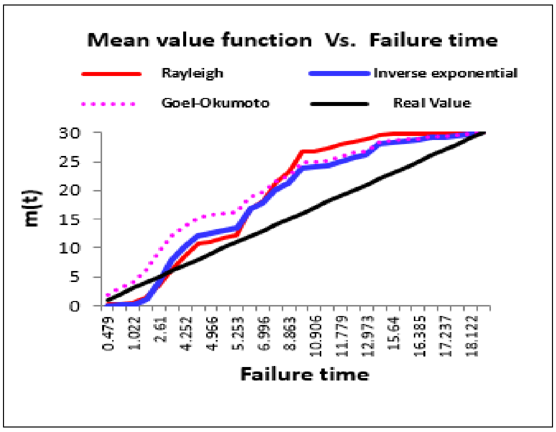

Fig 4. Pattern of Mean Value Function $m(t)$

Let us 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 $x_n = 18.735$, 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].

$$\hat{R}(\tau|x_n) = e^{-\int_{x_n}^{x_n+\tau} \lambda(\tau)d\tau} \quad (32)$$
$$= \exp[-\{m(\tau + x_n) - m(x_n)\}]$$

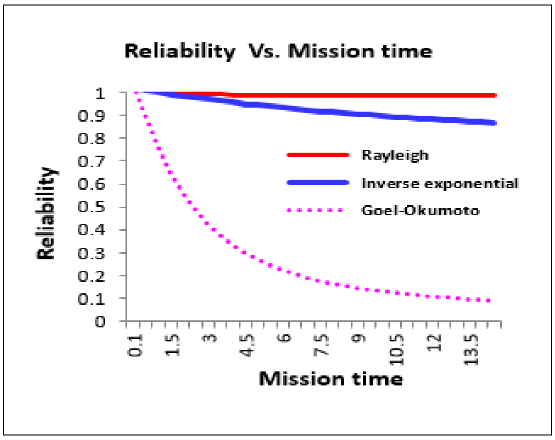

Fig 5. Transition of Reliability $\hat{R}(t)$

As shown in Fig 5, the Inverse-exponential model and the Rayleigh model show a higher reliability trend than the Goel-Okumoto model in which the reliability decreases with the mission time.

4. 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 compare and analyze the software reliability performance of the Inverse-exponential model and the Rayleigh model which is Weibull life distribution, 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, Inverse-exponential model was effective because the inverse-exponential model decreased along with the Rayleigh model as the failure time passed, and the mean square error (MSE) showed the smallest trend.

Second, in the performance analysis of the mean value function, all the proposed models showed overestimation patterns in the error estimation for true values, but the Inverse-exponential model was the most efficient because it had the smallest error value than the other models.

Third, in the performance analysis of mission reliability, the Inverse-exponential model and the Rayleigh model show stable and high reliability trends. On the other hand, Goel-Okumoto basic model shows that reliability decreases with the failure time. In other words, a comprehensive analysis of these simulations results shows that the inverse-exponential model is the most 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 basic design guideline.

In addition to, 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.

ACKNOWLEDGEMENT

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, pp. 85-121. 1999.

[2] Goel A L, Okumoto K, "Time-dependent fault detection rate model for software and other performance measures", IEEE Transactions on Software Engineering, Vol.28, pp. 206 -211, 1978.

[3] C. Y. Huang, "Performance analysis of software reliability growth models with testing-effort and change-point", Journal of Systems and Software, Vol.76, No.2, pp.181-194, 2005.

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

[5] K. S. Kim, H. C. Kim "The Comparative Study for Software Reliability Model Based on Finite and Infinite Failure Property using Rayleigh Distribution", Journal of Digital Con., Vol.12, No.12, pp.277-284, 2014.

[6] Pham H, Zhang X., “NHPP software re aliability and cost models with testing coverage", Eur.J. Oper. Res, 145, pp.445-454, 2003.

[7] Voda, R. GH., “On the inverse Rayleigh variable”, Rep.Stat. Apph. Res. Juse, vol. 19, No. 4, pp. 15-21, 1972.

[8] Ye Zhang, and Kaigui Wu, "Software Cost Model Considering Reliability and Time of Software in Use", Journal of Convergence Information Technology, Vol.7, No.13, pp.135-142, 2012.

[9] 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, Vol.7, No.983, pp.1-16, 2017.

[10] T. J. Yang, "The Comparative Study of NHPP Software Reliability Model Based on Log and Exponential Power Intensity Function", The Journal of Korea

Institute of Information, Electronics, and Communication Technology, Vol.8, No.6, pp.445-452, 2015.

[11] Hayakawa, Y. and Telfar, G., “Mixed poisson-type processes with application in software are reliability”, Mathematical and Computer Modelling, Vol. 31, pp.1 51-156, 2000.

[12] T. J. Yang, J.G. Park "A Comparative Study of the Software NHPP Based on Weibull Extension Distribution and Flexible Weibull Extension Distribution", International Journal of Soft Computing, Vol.11, No.4, pp276~281, 2016.

[13] T. J. Yang, "The Analysis and Predict of Software Failure Time Based on Nonlinear Regression", Journal of Engineering and Applied Sciences, Vol.13, Issue 12, pp.4376-4380, 2018.

[14] T. J. Yang, "A Comparative Study on Reliability Attributes of Software Reliability Model Based on Type-2 Gumbel and Erlang Life Distributionl", Journal of Engineering and Applied Sciences, Vol.14, Issue 10, pp.3366-3370, 2019.