Journal of Engineering and Applied Sciences 14 (10): 3366-3370, 2019
ISSN: 1816-949X
© Medwell Journals, 2019

A Comparative Study on Reliability Attributes of Software Reliability Model Based on Type-2 Gumbel and Erlang Life Distribution

Tae-Jin Yang
Department of Electronic Engineering, University of Namseoul, Cheonan, Republic of Korea

Abstract: With the rapid growth of the software industry, reliability research on software has the biggest impact on system reliability and software quality improvement. In this study, we compared and analyzed the attributes of the software reliability model dependent on the life distribution of the Type-2 Gumble model and the Erlang Model with Goel-Okumoto Basic Model. Software failure time data were used to identify software failure phenomena and the maximum likelihood estimation method was used for parameter estimation. As a result in terms of the intensity function, the Type-2 Gumble Model showed a tendency to decrease as the failure time progressed compared to the Goel-Okumoto Basic Model indicating that it was an efficient model but the Erlang Model increased inversely. In the pattern of the mean value function, the Type-2 Gumble Model showed some overestimation pattern compared to the true value but the error width was smaller than the Erlang Model. In addition as a result of comparing reliability by applying future mission time, the Type-2 Gumble Model was stable with the Goel-Okumoto Basic Model but the Erlang Model showed a decreasing tendency. Through this study, we have newly analyzed the attributes of software reliability on Type-2 Gumble Model without existing research examples and Erlang Model widely used in reliability field and it was able to present new research information that software developers could use as basic guidelines.

Key words: Reliability attributes, software reliability, finite failure NHPP, Type-2 Gumble life distribution, Erlang life distribution, Goel-Okumoto Basic Model

INTRODUCTION

With the rapid development of the information technology industry, the scale of computer software systems is constantly expanding and complicated.

For this reason, the study on software reliability has the greatest impact on system reliability and software quality improvement. Software reliability, defined as the probability that the software will function normally without failure for a specified amount of time in a particular environment is the most important software quality standard (Yamada and Osaki, 1985).

Over the past twenty years, many software reliability models based on the Non-Homogeneous Poisson Process (NHPP) have been proposed for software failure processes to predict software reliability. In other words, to estimate the reliability attributes such as the number of residual faults and the failure rate, a software reliability model based on NHPP using the defect intensity function and the mean value function in the controlled test environment was developed and proposed (Song *et al.*, 2017; Yang and Park, 2015). Yamada and Osaki (1985) stated that the results of the mean value function can be estimated using the maximum likelihood estimation method while Huang (2005) presented and explained a graph showing the confidence interval of the mean value function. In addition, Zhang and Wu (2012) proposed software reliability cost model based on software reliability model and software failure time.

In this study, based on the finite-fault NHPP Model, we have newly analyzed the attributes of the software reliability model based on the life distribution of the Type-2 Gumble Model without existing research examples and the Erlang Model widely used in the reliability field in addition to the Goel-Okumoto Basic Model and then present new research information through analysis of the proposed model.

MATERIALS AND METHODS

Related research

Type-2 Gumbel Model; Finite failure NHPP: The intensity function: λ(t) sand the mean value function m(t) of the NHPP whose life distribution follows the Type-2 Gumble distribution are as follows (Yang, 2015):

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

$$m(t|a, b) = \theta(e^{-bt^{-a}}) \qquad (2)$$

Note: $a, b>0, t\in[0, \infty]$

Therefore, using the the Eq. 1 and 2, the log-likelihood function of the finite-fault NHPP Model for parameter estimation can be derived as follows (Tae-Jin and Jea-Gun, 2016):

$$\begin{aligned}\mathrm{InL}_{\mathrm{NHPP}}(\Theta|x) &= \mathrm{nln}\theta+\mathrm{nlna}+\mathrm{nlnb}- \\ &(a+1)\sum_{i=1}^{n}\ln x_i - b\sum_{i=1}^{n} x_i^{-a-\theta e-bxn^{-a}}\end{aligned} \tag{3}$$

In this study, we consider the case of shape parameter a = 2. The partial derivatives of θ and b are as follows:

$$\frac{\partial \mathrm{InLNHPP}(\Theta|\underline{x})}{\partial\theta} = \frac{n}{\theta} - e^{-bx_n-2} = 0 \tag{4}$$

$$\begin{aligned}\frac{\partial \mathrm{InLNHPP}(\Theta|\underline{x})}{\partial\theta} &= \frac{n}{b} - \\ \sum_{i=1}^{n} x_i^{-2} &+ \theta x_n^{-2} e^{-bx_n^{-2}} = 0\end{aligned} \tag{5}$$

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ satisfying the following the Eq. 4 and 5 can be estimated by a numerical method.

Erlang Model; Finite failure NHPP: The Erlang distribution is the life distribution of the Gamma family widely used in the reliability field. The probability density function and the cumulative density function considering the shape parameter a and the scale parameter b are as follows (Goel and Okumoto, 1979; Yang, 2018):

$$f(t) = \frac{b^a}{\Gamma(a)} t^{a-1} e^{-bt} \tag{6}$$

$$F(t) = \left(1 - e^{-bt}\sum_{i=0}^{a-1}\frac{(bt)^i}{i!}\right) \tag{7}$$

Note: a, b>0, a = 1, 2, 2, ..., tϵ[0, ∞]. In this study, we consider the case of shape parameter a = 2. The log-likelihood function to Maximum Likelihood Estimation (MLE) by using the Eq. 6 and 7 is derived as follows (Tae-Jin and Jea-Gun, 2016).

$$\begin{aligned}&n\ln\theta - n\ln\Gamma(a)+\mathrm{nalnb}+(a\text{-}1)\sum_{i=1}^{n}\ln x_i - \\ &b\sum_{i=1}^{n} x_i - \theta+\theta e^{-bx_n}\left(\sum_{i=0}^{a-1}\frac{(bx_n)^i}{i!}\right)\end{aligned} \tag{8}$$

Note: x = (0≤x_1≤x_2≤, ..., ≤x_n, Θ is parameter space. The partial derivatives of θ and b are as follows:

$$\frac{\partial \mathrm{InLNHPP}(\Theta|\underline{x})}{\partial\theta} = \frac{n}{\theta} - 1+e^{-bx_n}(1+bx_n) = 0 \tag{9}$$

$$\frac{\partial \mathrm{InLNHPP}(\Theta|\underline{x})}{\partial\theta} = \frac{2n}{b} - \theta b x_n^{\,2} e^{-bx_n} - \sum_{i=1}^{n} x_i = 0 \tag{10}$$

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ satisfying the following the Eq. 9 and 10 can be estimated by a numerical method.

Goel-Okumoto Basic Model; Finite failure NHPP: The Goel-Okumoto Model is a well-known basic model in this field. Let f(t) and F(t) for the Goel-Okumoto Model (Goel and Okumoto, 1979) 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 fault strength function and the mean value function are as follows (Yamada and Osaki, 1985):

$$\lambda(t|\theta, b) = \theta f(t) = \theta b e^{-bt}, \ (\theta > 0,\ b > 0) > 0 \tag{11}$$

$$m(t|\theta, b) = \theta F(t) = \theta(1 - e^{-bt}) \tag{12}$$

Considering the failure time data up to the n^th^ and the Eq. 11 and 12, the likelihood function of the finite-fault NHPP Model is derived as follows (Song *et al*., 2017):

$$L(\theta, b|\underline{x}) = \left(\prod_{i=1}^{n}\theta b e^{-bxi}\right)\exp\left[-\theta(1 - e^{-bx_n})\right] \tag{13}$$

Note: x = (0≤x_1≤x_2≤, ..., ≤x_n. The likelihood function, using the Eq. 13 is simplified to the following log conditional expression:

$$\mathrm{InL}_{\mathrm{NHPP}}(\Theta|\underline{x}) = n\ln\theta+\mathrm{nlnb}\text{-}b\sum_{k=1}^{n} x_k - \theta(1 - e^{-bx_n}) \tag{14}$$

Therefore, the maximum likelihood estimators $\hat{\theta}_{MLE}$ and $\hat{b}_{MLE}$ for each parameter satisfy the following conditional expression:

$$\frac{n}{\hat{\theta}} = 1 - \exp(\_\hat{b}x_n) \tag{15}$$

$$\frac{n}{\hat{b}} = \sum_{i=1}^{n} x_n + \hat{\theta} x_n \exp(-\hat{b}x_n) \tag{16}$$

Table 1: Software failure time data

Failure No.	Failure time (h)	Failure interval-time	Failure time (h)×10^{-2}
1	30.02	30.02	0.30
2	31.46	1.44	0.31
3	53.93	22.47	0.53
4	55.29	1.36	0.55
5	58.72	3.43	0.58
6	71.92	13.20	0.71
7	77.07	5.15	0.77
8	80.90	3.83	0.80
9	101.90	21.00	1.01
10	114.87	12.97	1.14
11	115.34	0.47	1.15
12	121.57	6.23	1.21
13	124.97	3.40	1.24
14	134.07	9.10	1.34
15	136.25	2.18	1.36
16	151.78	15.53	1.51
17	177.50	25.72	1.77
18	180.29	2.79	1.80
19	182.21	1.92	1.82
20	186.34	4.13	1.86
21	256.81	70.47	2.56
22	273.88	17.07	2.73
23	277.87	3.99	2.77
24	453.93	176.06	4.53
25	535.00	81.07	5.35
26	537.27	2.27	5.37
27	552.90	15.63	5.52
28	673.68	120.78	6.73
29	704.49	30.81	7.04
30	738.68	34.19	7.38

Algorithm 1; The proposed analysis algorithm:

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

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

Step 2: Calculate the parameters $(\hat{\theta}, \hat{b})$ for the proposed model using the Maximum Likelihood Estimation

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

Step 4: Analyzes the transition ($m(t)$, $\lambda(t)$) of failure time for true values and reliability ($R(\tau)$) for future mission times

Step 5: Provide research information on the optimal model by analyzing the attributes of the proposed model

After analyzing the attributes of the proposed model using the above steps, we present information on the model that software developers need. In order to realize this process, we were assayed as follows:

Let compare and analyze the attributes of the proposed reliability models using the software failure time data (Prasad *et al*., 2011) (failure time information) as shown in Table 1. Laplace trend test was used to verify the reliability of the software failure time data (Yang, 2018).

RESULTS AND DISCUSSION

As a result of this test in this Fig. 1, the laplace factor between -2 and 2 appears confidence growing by reliability growth properties. Therefore, using this data, it is possible to estimate the time to failure time.

Table 2: Parameter estimation of each model

			Model comparison	
Models	MLE		MSE	R^2
Type-2 Gumbel	$\hat{\theta} = 30.3852$	$\hat{b} = 0.6960$	25.4057	0.9194
Erlang	$\hat{\theta} = 30.5978$	$\hat{b} = 0.0079$	45.4879	0.8558
Goel-Okumoto	$\hat{\theta} = 33.4092$	$\hat{b} = 0.3090$	5.84240	0.9814

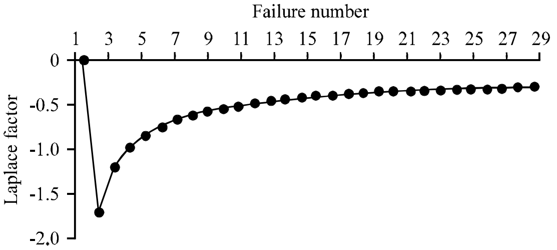

Fig. 1: Estimation results of Laplace trend

In this study, numerical conversion data (failure time [h]×10^{-2}) in order to facilitate the parameter estimation was used. These calculations, solving numerically, the initial values given to 0.001 and 3 and tolerance value for width of interval (10^{-5}) given using C-language checking adequate convergent were performed iteration of 100 times. And The Maximum Likelihood Estimation (MLE) was used to perform parameter estimation. The results of the parameter estimation are shown in Table 2.

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 (Chatterjee and Singh, 2014; Zhang and Wu, 2012):

$$\text{MSE} = \frac{\sum_{i=1}^{n}(m(x_i)-\hat{m}(x_i))^2}{n-k} \quad (17)$$

Note that, $m(x_i)$ is the total accumulated number of errors observed within time is $(0, x_i)$, $m(x_i)$ 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) can measure how successful the fit is in explaining the variation of the data. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(m(x_i)-\hat{m}(x_i))^2}{\sum_{i=1}^{n}(m(x_i)-\sum_{j=1}^{n}m(x_j)/n))^2} \quad (18)$$

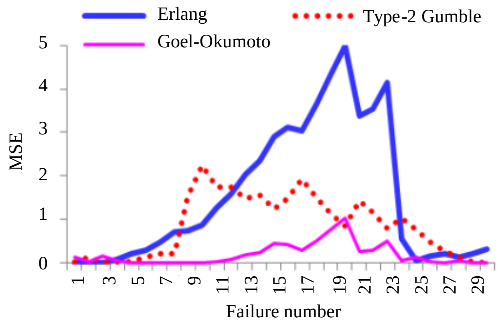

Fig. 2: Transition of mean square error

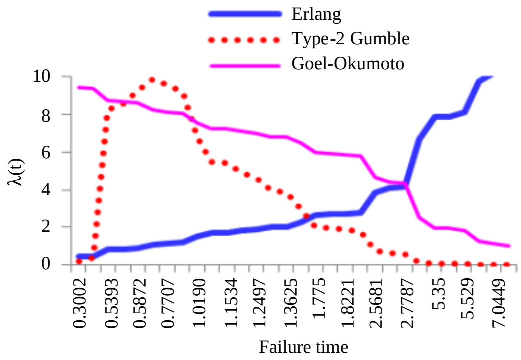

Fig. 3: Transition of intensity function

If the coefficient of determination is large in comparison, the error is relatively small and becomes an efficient model (Yang, 2017). As shown in Table 2, we can see that the Type-2 Gumble Model is more efficient than the Erlang 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.

Also, Fig. 2 shows the transition of Mean Square Error (MSE) according to each failure number. That is in this figure, the Type-2 Gumble Model shows better estimates than the Erlang distribution model in the total number of faults. In this figure, similar to the Goel-Okumoto Model, the mean squared error of the Type-2 Gumble Model shows a trend of decreasing with time which is more efficient than the Erlang Model in terms of fitness.

Figure 3 shows trends in the strength function which is the instantaneous failure rate. The Type-2 Gumble Model shows the greatest decreasing tendency as the failure time passes indicating that it is an efficient model but only the Erlang Model shows an increasing pattern.

Figure 4 shows the pattern trend for the mean value function. In this figure, the Erlang Model has underestimated significantly from the difference between the true values but the Type-2 Gumble Model has a slightly overestimated pattern from the difference between the true values. That is, the Type-2 Gumble Model is more efficient than the Erlang Model because the error width is estimated to be small.

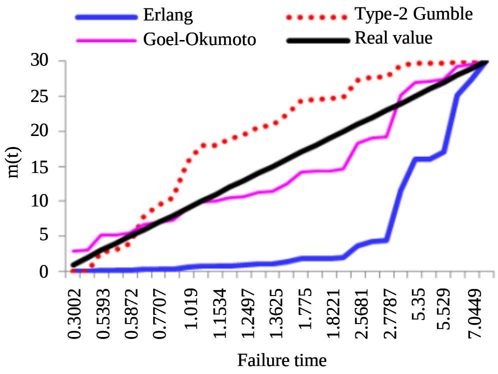

Fig. 4: Pattern of mean value function

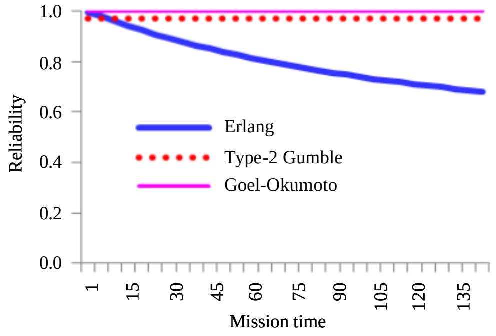

Fig. 5: Transition of reliability

A software failure occurred at the time of testing x_n = 7.3868 of the NHPP Model and which is the probability that a software failure does not occur between confidence interval (x_n, $x_n+\tau$) (where τ is the mission time at future), reliability is derived from the Eq. 19 as follows (Chatterjee and Singh, 2014; Zhang and Wu, 2012):

$$\hat{R}(\tau|x_n) = e - \int_{x_n}^{x_n+\tau}\lambda(\tau)d\tau = \exp[-\{m(\tau+x_n) - m(x_n)\}] \quad (19)$$

As shown in Fig. 5, the Type-2 Gumble Model and the Goel-Okumoto Model show a higher reliability trend than the Erlang Model in which the reliability decreases with the mission time in the future.

CONCLUSION

After quantitatively modeling the attributes of failure or failure occurrence phenomenon during software testing or actual software use, analyzing the usefulness of software can provide a relatively efficient reliability assessment. In this study, based on the finite-fault NHPP Model with software failure time data, we compare and analyze the software reliability model of the Type-2 Gumbel Model and the Erlang Model which depends on the lifetime distribution, together with Goel-Okumoto Basic Model.

The results of this study can be summarized as follows: first in terms of intensity function, the Type-2 Gumble Model shows the most efficient model as the failure time decreases as compared to the Goel-Okumoto Basic Model but the Erlang Model increases inversely. Second in the pattern of the averaged value function, the Erlang Model underestimates the error from the true value and the Type-2 Gumble Model shows a slight overestimation pattern. However, it is found that the Type-2 Gumble Model is more efficient than the Erlang Model because of the small error margin.

Third in the reliability, the Type-2 Gumble Model and the Goel-Okumoto Basic Model show stable and high reliability trends. On the other hand, Eralng Model shows that reliability decreases with the failure time. As a result, through this study, along with a new analysis on the attributes 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.

ACKNOWLEDGEMENT

Funding for this study was provided by Namseoul University.

REFERENCES

Chatterjee, S. and J.B. Singh, 2014. A NHPP based software reliability model and optimal release policy with logistic-exponential test coverage under imperfect debugging. Intl. J. Syst. Assur. Eng. Manage., 5: 399-406.

Goel, A.L. and K. Okumoto, 1979. Time-dependent error-detection rate model for software reliability and other performance measures. IEEE. Trans. Reliab., 28: 206-211.

Huang, C.Y., 2005. Performance analysis of software reliability growth models with testing-effort and change-point. J. Syst. Software, 76: 181-194.

Prasad, R.S., K. Ramchand, H. Rao and R.R.L. Kantham, 2011. Software reliability measuring using modified maximum likelihood estimation and SPC. Int. J. Comput. Applic., 21: 1-5.

Song, K.Y., I.H. Chang and H. Pham, 2017. A software reliability model with a Weibull fault detection rate function subject to operating environments. Appl. Sci., 7: 1-16.

Tae-Jin, Y. and P. Jea-Gun, 2016. The comparative study of the software NHPP based on weibull extension distribution and flexible weibull extension distribution. Intl. J. Soft Comput., 11: 276-281.

Yamada, S. and S. Osaki, 1985. Software reliability growth modeling: Models and applications. IEEE. Trans. Software Eng., 11: 1431-1437.

Yang, T.J. and J.G. Park, 2015. A study on software reliability model based on mixture Weibull NHPP property. Intl. J. Appl. Eng. Res., 10: 548-552.

Yang, T.J., 2015. The comparative study of NHPP software reliability model based on log and exponential power intensity function. J. Korea Instit. Inf. Electron. Commun. Technol., 8: 445-452.

Yang, T.J., 2017. The performance analysis comparative study depend on software reliability model and curve regression model. Intl. J. Soft Comput., 12: 313-317.

Yang, T.J., 2018. The analysis and predict of software failure time based on nonlinear regression. J. Eng. Appl. Sci., 13: 4376-4380.

Zhang, Y. and K. Wu, 2012. Software cost model considering reliability and time of software in use. J. Convergence Inform. Technol., 7: 135-142.