

A Comparative Study on the Cost and Release Time of Software Development Model Based on Lindley-Type Distribution

Tae-Jin Yang¹

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

ORCID: 0000-0001-7971-277X

Abstract

In this study, the software development cost model with Lindley-Type (Basic-Lindley and Modified-Lindley) distribution property was compared with the Goel-Okumoto basic model, and the attributes of software development cost and optimal release time were newly analyzed. For this purpose, software failure time data were used, parameter estimation was applied to the maximum likelihood estimation method, and nonlinear equations were calculated using the bisection method. As a result of comparing Lindley-Type models, we confirmed that the Modified-Lindley model is more efficient because it has lower software development costs and faster software release time than the Basic-Lindley model. Also, it was confirmed that the Goel-Okumoto basic model is relatively efficient than the Lindley-Type model. Through this study, we newly analyzed the attributes of the Lindley-Type software development cost model which has no previous study case. Also, it is expected that software developers will be able to use this study data as a basic guideline for exploring the attributes of economical software development cost and optimal release time.

Keywords: Basic-Lindley Distribution, Goel-Okumoto Basic Model, Modified-Lindley Distribution, NHPP, Software Release Time, Software Development Cost.

1. INTRODUCTION

Software technology, which is the core of the 4th industrial revolution era, has converged rapidly in various industrial fields, and the necessity for software development that can process a large amount of data accurately without failure is also increasing. In this software development process, if the economic software development cost and the optimal release time can be predicted, the efficient development process can be performed. For this reason, researches on software development costs have been actively conducted along with software reliability issues. In this process, software reliability models and software development cost models using the Non-Homogeneous Poisson Process (NHPP) have been proposed [1]. In particular, concerning the NHPP reliability model, Goel and Okumoto [2] proposed an exponential software

reliability model, Shyur [3] proposed a generalized reliability model using change-point. Pham and Zhang [4] proposed a new model based on the NHPP software reliability with testing coverage, and Kim [5] analyzed the cost of the software development model based on the Burr-Hatke exponential distribution. Also, Yang [6] proposed the analysis and prediction methods of the software reliability model based on the NHPP software reliability model with Logistic Distribution Property. Therefore, in this study, after the software development cost model with Lindley-Type (Basic-Lindley and Modified Lindley) lifetime distribution property is comparing with the Goel-Okumoto basic model, we analyze the attributes of development cost and release time through the research results, and will present the information on economic development cost and optimal release time.

2. RELATED RESEARCH

2.1 Goel-Okumoto Basic Model

The Goel-Okumoto model is a well-known basic model in the software reliability field. This model assumes the exponential distribution as the lifetime distribution per fault. Let $f(t)$ and $F(t)$ for the Goel-Okumoto basic model be a probability density function and a cumulative density function, respectively. Assuming that the expected value of the defect that can be observed up to the observation point $[0, t]$ is θ , the strength function $\lambda(t)$ and the mean value function $m(t)$ are as follows.

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

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

Note that $t \in [0, \infty]$ and $b > 0$ are the shape parameter.

The likelihood function of the finite-fault NHPP model using Eq. 1 and Eq. 2 is derived as follows. Here, the finite failure means that no new defect occurs during the repair period.

$$L(\theta, b|\underline{x}) = \left(\prod_{i=1}^n \theta b e^{-bx_i} \right) \exp[-\theta(1 - e^{-bx_n})] \quad (3)$$

The log-likelihood function of the finite-fault NHPP model by using Eq. 3 is derived as follows.

$$\ln L_{NHPP}(\theta|\underline{x}) = n \ln \theta + n \ln b - b \sum_{i=1}^n x_i - \theta(1 - e^{-bx_n}) \quad (4)$$

Note that $\underline{x} = (0 \leq x_1 \leq x_2 \leq \dots \leq x_n)$,
 $\theta = \{\theta, b\}$ specifies parameter space.

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and \hat{b}_{MLE} satisfying the following Eq. 5 and Eq. 6 can be estimated by a numerical method.

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - 1 + e^{-\hat{b}x_n} = 0 \quad (5)$$

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial b} = \frac{n}{\hat{b}} - \sum_{i=1}^n x_n - \hat{\theta} x_n e^{-\hat{b}x_n} = 0 \quad (6)$$

2.2 Basic-Lindley Distribution NHPP Model

The Basic-Lindley life distribution is widely known as a suitable model for lifetime test and stress-strength reliability. The Basic-Lindley distribution is a mixture type of exponential distributions and gamma distributions [7]. Let $f(t)$ and $F(t)$ for the Basic-Lindley model be a probability density function and a cumulative density function, respectively. The probability density function and the cumulative density function are as follows.

$$f(t) = \frac{b^2}{b+1} (1+t) \times e^{-bt} \quad (7)$$

$$F(t) = 1 - \left[\left(\frac{b+1+bt}{b+1} \right) \times e^{-bt} \right] \quad (8)$$

Note that $t \in [0, \infty]$ and $b > 0$ are the shape parameter.

Assuming that the expected value of the defect that can be observed up to the observation point $[0, t]$ is θ , the finite fault strength function $\lambda(t)$ and the mean value function $m(t)$ are as follows.

$$\lambda(t|\theta, b) = \theta f(t) = \theta \left[\frac{b^2}{b+1} (1+t) \times e^{-bt} \right] \quad (9)$$

$$m(t|\theta, b) = \theta F(t) = \theta \left[1 - \left(\frac{b+1+bt}{b+1} \right) \times e^{-bt} \right] \quad (10)$$

The likelihood function of the finite-fault NHPP model by using Eq. 9 and Eq. 10 is derived as follows.

$$L_{NHPP}(\theta|\underline{x}) = \left[\prod_{i=1}^n \theta \left[\frac{b^2}{b+1} (1+t) \times e^{-bt} \right] \right] \cdot \exp \left[-\theta \left[1 - \left(\frac{b+1+bt}{b+1} \right) \times e^{-bt} \right] \right] \quad (11)$$

The log-likelihood function of the finite-fault NHPP model by using Eq. 11 is derived as follows.

$$\ln L_{NHPP}(\theta|\underline{x}) = -\theta \left[1 - \left(\frac{b+1+bt}{b+1} \right) \times e^{-bt} \right] + n \ln \theta + 2n \ln b - n \ln(b+1) + \sum_{i=1}^n (1+x_i) - b \sum_{i=1}^n x_i \quad (12)$$

Therefore, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and \hat{b}_{MLE} satisfying the following Eq. 13 and Eq. 14 can be estimated by a numerical method.

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial \theta} = \frac{n}{\hat{\theta}} - \left[1 - \left(\frac{b+1+bt}{b+1} \right) \times e^{-bt} \right] = 0 \quad (13)$$

$$\frac{\partial \ln L_{NHPP}(\theta|\underline{x})}{\partial b} = \frac{2n}{b} - \frac{n}{b+1} - \sum_{i=1}^n x_i - \theta e^{-bx_n} (x_n - b^2 x_n^2 + b - b^3 x_n^3 - b^3) = 0 \quad (14)$$

2.3 Modified-Lindley Distribution NHPP Model

Shanker, R [8] proposed a Modified-Lindley model that modified the Basic-Lindley model. Let $f(t)$ and $F(t)$ for the Modified-Lindley model be a probability density function and a cumulative density function, respectively. The probability density function and the cumulative density function are as follows.

$$f(t) = \frac{b^2}{b^2+1} (b+t) \times e^{-bt} \quad (15)$$

$$F(t) = 1 - \left[\left(\frac{b^2+1+bt}{b^2+1} \right) \times e^{-bt} \right] \quad (16)$$

Note that $t \in [0, \infty]$ and $b > 0$ are the shape parameter.

Assuming that the expected value of the defect that can be observed up to the observation point $[0, t]$ is θ , the finite fault strength function $\lambda(t)$ and the mean value function $m(t)$ are as follows.

$$\lambda(t|\theta, b) = \theta f(t) = \theta \left[\frac{b^2}{b^2+1} (b+t) \times e^{-bt} \right] \quad (17)$$

$$m(t|\theta, b) = \theta F(t) = \theta \left[1 - \left(\frac{b^2+1+bt}{b^2+1} \right) \times e^{-bt} \right] \quad (18)$$

The likelihood function of the finite-fault NHPP model by using Eq. 17 and Eq. 18 is derived as follows.

$$L_{NHPP}(\theta|\underline{x}) = \left[\prod_{i=1}^n \theta \frac{b^2}{b^2+1} (b+t) e^{-bt} \right] \cdot \exp \left[-\theta \left(1 - \frac{b^2+1+bt}{b^2+1} e^{-bt} \right) \right] \quad (19)$$

After solving the log-likelihood function by using the Eq. 19, the maximum likelihood estimator $\hat{\theta}_{MLE}$ and \hat{b}_{MLE} satisfying the following Eq. 20 and Eq. 21 can be estimated by a numerical method.

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - \left[1 - \left(\frac{b^2 + 1 + bt}{b^2 + 1} \right) \times e^{-bt} \right] = 0 \quad (20)$$

$$\frac{\partial \ln L_{NHPP}(\theta | \underline{x})}{\partial b} = -\theta x_n e^{-bx_n} \left[\frac{(b^4 + b^3 x_n + 3b^2 + bx_n)}{(b^2 + 1)^2} \right] + \frac{2n}{b} - \frac{2nb}{b^2 + 1} + \sum_{i=1}^n \frac{1}{b + x_i} - \sum_{i=1}^n x_i = 0 \quad (21)$$

2.4 Software Development Cost Model

The estimated total cost of development based on the software development cost model is as follows [9].

$$E_t = E_1 + E_2 + E_3 + E_4 \quad (22)$$

$$= E_1 + C_2 \times t + C_3 \times m(t) + C_4 \times [m(t + t') - m(t)]$$

Note that E_t is the estimated total cost of software development.

① E_1 stands for initial software development costs, and is considered a constant.

② E_2 is the testing cost per unit time, and the actual cost per unit time is different for each industry section. E_2 is expressed by the following Eq. 23.

$$E_2 = C_2 \times t \quad (23)$$

Note that C_2 is the testing cost per unit time, t is the testing time point.

③ E_3 represents the cost of removing a defect by detecting a basic defect and is expressed by the following Eq. 24.

$$E_3 = C_3 \times m(t) \quad (24)$$

Note that C_3 is the cost of removing one defect found in the testing process, and the mean value function $m(t)$ is the expected value of the defect that can be detected at time t .

④ E_4 represents the cost of eliminating all remaining defects in the software operating system and is expressed by the following Eq. 25.

$$E_4 = C_4 \times [m(t + t') - m(t)] \quad (25)$$

Note that C_4 is the defect correction cost discovered by the operator at the software operation stage after the software is released, and t' is the time when the software can be operated normally after releasing the software.

In reality, C_4 has a higher cost than C_2 and C_3 . Therefore, this study is applied to the real situation that C_4 is higher than C_2 and C_3 .

Therefore, the optimal software release time for the software development cost can be derived as follows.

$$\frac{\partial E_t}{\partial t} = E' = (E_1 + E_2 + E_3 + E_4)' = 0 \quad (26)$$

In other words, it can be seen that the optimal release time is the time point at which the lowest development cost is satisfied.

3. THE PROPOSED ALGORITHM AND SOLUTIONS

In this study, the proposed algorithm for analyzing and predicting software development costs is as follows.

Software development cost analysis algorithm:

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

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

Step 3: Analyzing the results of changing the costs (C_3 , C_4) that make up the total cost of software development.

Step 4: Determining the best efficient model that meets both optimal software development cost and release time.

Step 5: Providing the analysis information about software development cost and release time with reliability verification.

Let compare and analyze the attributes of the proposed development cost models using the software failure time data [10] as shown in Table 1. This software failure is the data that was occurred 30 times in 187.35 unit time.

Table 1. Software Failure Time Data

Failure Number	Failure Time (hours)	Failure Number	Failure Time (hours)
1	4.79	16	107.71
2	7.45	17	109.06
3	10.22	18	111.83
4	15.76	19	117.79
5	26.10	20	125.36
6	35.59	21	129.73
7	42.52	22	152.03
8	48.49	23	156.40
9	49.66	24	159.80
10	51.36	25	163.85
11	52.53	26	169.60
12	65.27	27	172.37
13	69.96	28	176.00
14	81.70	29	181.22
15	88.63	30	187.35

Laplace trend test was used to verify the reliability of the software failure time data as shown in Fig 1 [11].

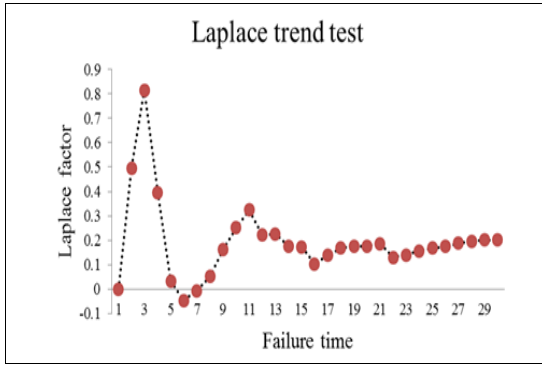


Fig 1. Estimation results of Laplace trend test

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 Figure 1, 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.

In this study, the Maximum Likelihood Estimation (MLE) was used to perform parameter estimation [12]. And numerical conversion data (Failure time[hours] $\times 10^{-1}$) 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	
	$\hat{\theta}_{MLE}$	\hat{b}_{MLE}
Goel-Okumoto	32.9261	1.297×10^{-1}
Basic-Lindley	37.8877	1.497×10^{-1}
Modified-Lindley	36.0896	1.703×10^{-1}

Explanatory notes. MLE: Maximum Likelihood Estimation.

In this study, we assumed the cost of software development as [Supposition 1] ~ [Supposition 4] to simulate the same as the actual software development environment. To do this, we will analyze and predict software development cost and release time by changing each component (C_3, C_4, t') that constitutes the total software development cost [13].

[Supposition 1: Basic conditions]

$$E_1 = 40\$, C_2 = 5\$, C_3 = 1.5\$, C_4 = 20\$, t' = 20 \quad (27)$$

The result of the cost curve using [Supposition 1] is as shown in Figure 2. In this figure, the transition of the development cost curve shows a constant pattern for a short period after showing a decreasing pattern in the initial stage, but it shows an increasing pattern as the release time passes. The reason is that in the process of eliminating defects during the initial stage, the development cost is decreased because the number of defects inherent in the software is reduced. But the development cost is increased because the probability of finding the remaining defects during the latter stage is gradually lowered. As a result, the pattern of the development cost curve gradually increases as the release time passes.

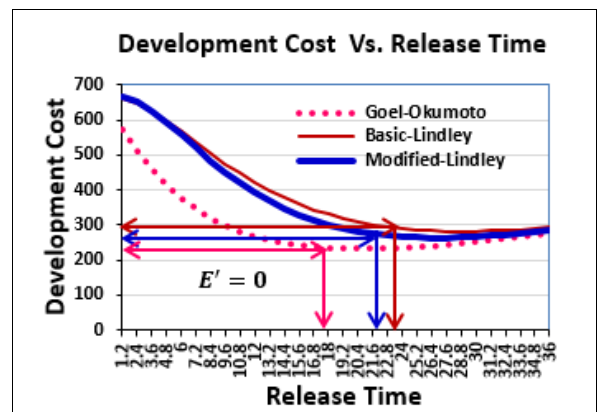


Fig 2. The development cost curve applied to the condition of [Supposition 1]

The simulation results shown in Figure 2 show that the development cost of the Goel-Okumoto basic model is 230\$ and the release time is 17.8H. And the development cost of the Modified-Lindley model is 270\$ and the release time is 21.8H. The development cost of the Basic-Lindley model is 300\$ and the release time is 23.4H. Although all of the proposed models show a similar pattern, the Goel-Okumoto basic model is the best efficient because it has lower development costs and faster release time than other models (Modified-Lindley and Basic-Lindley).

[Supposition 2: Assumed that the cost C_3 is increased in supposition 1]

$$E_1 = 40\$, C_2 = 5\$, C_3 = 3\$, C_4 = 20\$, t' = 20 \quad (28)$$

[Supposition 2] is a case where the cost (C_3) of removing one defect found in the software testing process is doubled compared to [Supposition 1]. The simulation results are shown in Fig. 3 show that the development cost of the Goel-Okumoto basic model is 290\$ and the release time is 17.8H. And the development cost of the Modified-Lindley model is

310\$ and the release time is 21.8H. The development cost of the Basic-Lindley model is 340\$ and the release time is 23.4H.

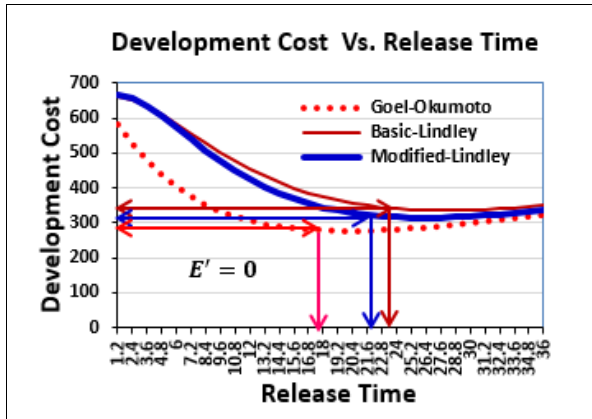


Fig 3. The development cost curve applied to the condition of [Supposition 2]

In other words, the development cost has increased, but the release time has not changed. Therefore, in this case, as many defects as possible should be removed at once so that the cost of removing one defect in the software testing step is not increased. Also, the Modified-Lindley model is relatively superior to the Basic-Lindley model because of its lower development cost and faster release time.

[Supposition 3: Assumed that the cost C_4 is increased in supposition 1]

$$E_1 = 40$, $C_2 = 5$, $C_3 = 1.5$, $C_4 = 40$, $t' = 20$ (29)$$$$$

[Supposition 3] is a case where the cost(C_4) of correcting defects discovered by the software operator during the operation stage after releasing the software is doubled compared to [Supposition 1].

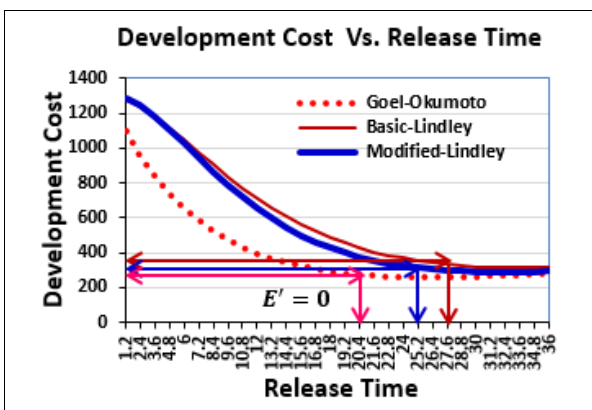


Fig 4. The development cost curve applied to the condition of [Supposition 3]

As shown in Figure 4, it can be seen that the release time is delayed with increasing development costs. Therefore, in this case, we must eliminate all possible defects at the testing stage rather than the operational stage to reduce all defects before releasing the software. Also, the Goel-Okumoto basic model is the best efficient because it has lower development costs and faster release time than the Lindley-Type model. If comparing the Lindley-Type models by the same method, we can see that the Modified-Lindley model is relatively superior to the Basic-Lindley model.

In conclusion, we can predict the optimal software release time together with the development cost trend.

[Supposition 4: Assumed that the time t' is increased in supposition 1]

$$E_1 = 40$, $C_2 = 5$, $C_3 = 1.5$, $C_4 = 20$, $t' = 40$ (30)$$$$$

[Supposition 4] is a case where the time(t') that the software can be operated normally after releasing the software is doubled compared to [Supposition 1]. As shown in Figure 5, the development cost has increased, but the release time has not changed. Also, in this figure, before the optimal release time, as the time of t' increases, the cost increases, but after the optimal release time for all models, the cost is almost the same.

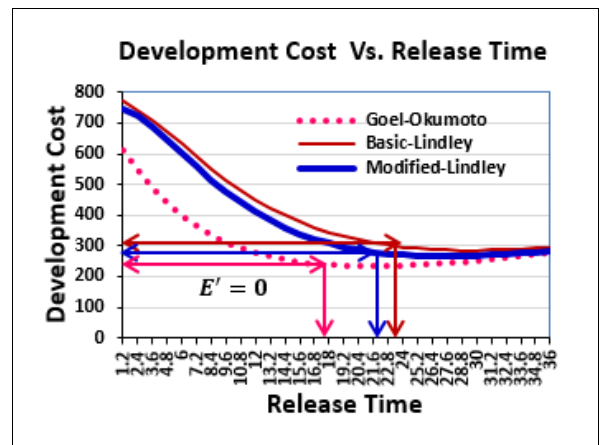


Fig 5. The development cost curve applied to the condition of [Supposition 4]

Therefore, we can analyze the optimal software release time together with the development cost trend, and also, software operators are considered to be helpful to predict the software development costs and the release time [14].

4. CONCLUSION

If software development costs can be quantitatively modeled together with release time during the software development

process, the attributes of development costs can be efficiently analyzed and predicted. Therefore, this study analyzes and predicts software development cost together with optimal software release time through the NHPP reliability models based on Lindley-Type distribution property.

The results of this study can be summarized as follows.

First, under the given basic conditions (Supposition 1), the software development cost curve shows a constant pattern for a short time after a significant decrease in the initial stage but shows a pattern of increasing again in the latter stage as the release time passes.

Second, before software release, if the cost (C_3) of removing one defect found in the testing process increases, the development cost has increased as well, but the release time has not changed. However, after the software release, if the defect correction cost (C_4) discovered by the software operator increases, the development cost increases, and the release time is also delayed.

Third, before the optimal release time, as the time (t') when the software can be operated normally after releasing the software system increases, the cost increases, but after the optimal release time, the cost is almost the same for all models.

As a result of a comprehensive analysis of the proposed model used in this study, the Goel-Okumoto basic model is relatively efficient because it has low software development cost and fast release time compared to the Lindley-Type model. Also, when comparing the Lindley-Type model, it can be seen that the Modified-Lindley model is relatively superior to the Basic-Lindley model.

Using the results of this study, it is possible to provide software developers and operators with the necessary prior information for predicting the most economical software development costs and the optimal release time. Also, further studies will be needed to find out the optimal software development cost model through analysis with other models having the same type of failure time data distribution.

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, pp. 5-121, 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] Shyur H-J, "A stochastic software reliability model with imperfect debugging and change-point," *J. Syst. Software* 66, pp. 135-141, 2003.
[4] Pham H, Zhang X., "NHPP software reliability and cost models with testing coverage," *Eur.J. Oper. Res.*,145, pp. 445-454, 2003.
[5] Hee-Cheul Kim, "A Comparative Study on the Cost of Software Development Model Based on Burr-Hatke Exponential Distribution," *International Journal of Engineering Research and Technology*, Vol.12, No. 11, pp. 2036-2040, 2019.
[6] T. J. Yang, "A Comparative Study on the Performance Attributes of Finite Failure NHPP Software Reliability Model with Logistic Distribution Property", *International Journal of Engineering Research and Technology*, Vol.13, No.3, pp. 438-443, 2020.
[7] Adhikari, T. R. and Srivastava, R.S., "Poisson-Size biased Lindley Distribution," *International Journal of Scientific and Research Publications*, Vol. 4, No. 1, pp. 1-6, 2014.
[8] Shanker, R., Shanker., "distribution and Its Applications," *International Journal of Statistics and Applications*, Vol. 5, No. 6, pp. 338-348, 2015.
[9] 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.
[10] Hayakawa, Y. and Telfar, G., "Mixed Poisson-type processes with application in software reliability," *Mathematical and Computer Modelling*, Vol. 31, pp. 151-156, 2000.
[11] Tae-Jin 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.
[12] Hee-Cheul Kim, "A 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*, Vol. 12, No. 1, pp. 73-78, 2019.
[13] 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*, Vol. 12, No. 11, pp. 1890-1896, 2019.
[14] 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*. Vol. 11, No. 4, pp. 276-281, 2016.