American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Improving Software Reliability Prediction Through Incorporating Learning Effects

Lutfiah Ismail Al turk*

Statistics Department, Faculty of Sciences, King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia
Email: lturk@kau.edu.sa

Abstract

Software reliability is one of the major metrics for software quality evaluation. In reliability engineering, testing phase specifying the process of measuring software reliability. In this paper; we examine the effect of incorporating the autonomous errors detected factor and learning factor in enhancing the prediction accuracy with application to software failure data. For this purpose, Non-Homogenous Poisson Process (NHPP) model with the perspective of learning effects based on the Log-Logistic (LL) distribution is proposed. The parameter estimation using the Non-Linear Least Squares Estimation (NLSE) method is conducted. Two goodness-of-fit tests are used to evaluate the proposed models. This paper encourages software developers to consider the learning effects property in software reliability modeling.

Keywords: Non-homogeneous Poisson process; log-logistic distribution; learning effects; goodness-of-fit performance; non-linear least squares estimation.

1. Introduction

A recently settled software system prior to its use is exposed to a robust testing in order to reduce the likelihood of failure manifestation and guarantee that the system will behave as expected. Software solutions for critical application fields demand a much intensive amount of testing. Software Reliability Growth Models (SRGMs) are the models that attempt to predict software reliability using data from testing phase. Over the years, many SRGMs that belong to the Non-Homogeneous Poisson Process (NHPP) have been suggested. Many researchers aim to better describe the failure phenomena by incorporating some representative factors to these models [1]-[4]. Learning effects perspective can be stated to predict failures that are expected to occur in specified operations, recognizing spots of which faults that need the most efforts to be fixed. Several NHPP SRGMs have been enhanced by incorporating learning effects [5]-[7].

----

* Corresponding author.

In this article, NHPP SRGM with the concept of learning effect based on the Log-Logistic (LL) distribution is proposed. The Non-Linear Least Squares Estimation (NLSE) method is used for the proposed model parameter estimation. The validity of the proposed models is illustrated through six software failure data sets. The Mean Square Error (MSE) and coefficient of determination criteria R^2 are employed for model prediction accuracy. The rest of the paper is arranged as follows: Section 2 introduces the NHPP models, the NHPP LL model, and the improved NHPP LL model. Section 3 discusses parameter estimation using the NLSE method. Section 4 presents the selected model evaluation criteria. Section 5 illustrates the results and offers comparative analysis on the basis of six real data sets. Section 6 presents the conclusion.

2. The NHPP Models

The NHPP models assist to express the failure occurrence during the testing phase of a software system, $\mathrm{N(t_i)}$ which represents the cumulative number of errors detected by time $\mathrm{t_i}$; $\mathrm{i} = (1, 2, \dots, \mathrm{n})$ follows the Poisson distribution as follows:

$$\mathrm{P(N(t_i) = k)} = \frac{[\mathrm{m(t_i)}]^k}{\mathrm{k!}} \mathrm{e}^{-\mathrm{m(t_i)}}, \quad \text{where } \mathrm{k} = 0,1,\dots \quad , \tag{1}$$

$\mathrm{m(t_i)}$ indicating the expected number of errors found within time (0, $\mathrm{t_i}$) and described by:

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

where $\lambda(s)$ is the intensity function which conversely can be written as:

$$\lambda(\mathrm{t_i}) = \frac{\mathrm{dm(t_i)}}{\mathrm{dt}}. \tag{3}$$

Any NHPP model can be defined completely by knowing either its mean value function or intensity function. This type of modeling has several reliability characteristics among them the Mean Time Between Failure (MTBF) which assesses the length of time that a software system remains in operation. The instantaneous and cumulative MTBF can be respectively given as:

$$\mathrm{MTBF}_I(\mathrm{t_i}) = \frac{1}{\lambda(\mathrm{t_i})}. \tag{4}$$

$$\mathrm{MTBF}_c(\mathrm{t_i}) = \frac{\mathrm{t_i}}{\mathrm{m(t_i)}}. \tag{5}$$

Some of the essential assumptions of the NHPP models are:

1. The failure phenomenon is modeled by the NHPP.
2. During execution of a software system, failure occurrence is caused by faults latent in the system.
3. The number of detected errors is proportional to the number of remaining errors in the software system.
4. Faults remaining in the software system evenly affect the software failure rate.
5. Once a failure occurs, the causing faults are detected and removed with certainty.

6. All software faults are mutually independent.

2.1. NHPP Log-Logistic (LL) Model

The Probability Density Function (PDF) of the Log-Logistic (LL) distribution is:

$$f(\mathrm{t_i}) = \frac{\mathrm{ab}t_i^{b-1}}{(1+at_i^b)^2}. \tag{6}$$

And the corresponding Cumulative Distribution Function (CDF) is:

$$\mathrm{F(t_i)} = \frac{at_i^b}{1+at_i^b}. \tag{7}$$

While its hazard function is defined as follows:

$$\mathrm{h(t_i)} = \frac{f(\mathrm{t_i})}{1-\mathrm{F(t_i)}}$$

$$= \frac{\mathrm{ab}t_i^{b-1}}{1+at_i^b}, \tag{8}$$

where $a, b > 0, a$ is positive scale parameter, and b is shape parameter. The mean value function of the NHPP LL model is given as follows [8]:

$$\mathrm{m(t_i)} = \theta\, \mathrm{F(t_i)}$$

$$= \frac{\theta a t_i^b}{1+at_i^b}. \tag{9}$$

where $\mathrm{t_i},\ \mathrm{i} = (1, 2, \ldots, \mathrm{n})$ is the failure times, $\theta > 0$ is the number of initial errors, whereas the failure intensity function is defined as:

$$\lambda(\mathrm{t_i}) = \grave{\mathrm{m}}(\mathrm{t_i})$$

$$= \frac{\theta abt_i^{b-1}}{(1+at_i^b)^2}. \tag{10}$$

2.2. Modified NHPP LL Model

In this section the NHPP LL model will be enhanced by incorporating the autonomous errors detected factor and learning factor for discovering the software faults in a system. Certainly, the efficiency in terms of software debugging can be enhanced based on these two factors. Following the work of [9] the CDF can be modified as follows:

$$\mathrm{F}(\mathrm{t_i}) = \frac{\mathrm{h}(\mathrm{t_i})-\eta}{\gamma} \; , \tag{11}$$

where $\eta > 0$ is the autonomous errors-detected factor and $\gamma > 0$ is the learning factor. Hence, the CDF of the NHPP LL model can be rewritten as follows:

$$\mathrm{F}(\mathrm{t_i}) = \frac{\frac{\mathrm{ab}t_i^{b-1}}{1+at_i^b}-\eta}{\gamma}$$

$$= \frac{\mathrm{ab}t_i^{b-1}-\eta(1+at_i^b)}{\gamma(1+at_i^b)}. \tag{12}$$

Then, the mean value function of the modified NHPP LL model is given as follows:

$$\mathrm{m}(\mathrm{t_i}) = \theta\, \mathrm{F}(\mathrm{t_i})$$

$$= \frac{\theta\left(\mathrm{ab}t_i^{b-1}-\eta(1+at_i^b)\right)}{\gamma(1+at_i^b)}. \tag{13}$$

And the failure intensity function is defined as:

$$\lambda(\mathrm{t_i}) = \theta\, \grave{\mathrm{F}}(\mathrm{t_i})$$

$$= \frac{\theta(\mathrm{ab}t_i^{b-2}-\left(\mathrm{b}-1-at_i^b\right))}{\gamma(1+at_i^b)^2}. \tag{14}$$

Another two reliability characteristics of the modified NHPP LL are: the instantaneous MTBF, which can be obtained using Eq. (4) as follows:

$$\mathrm{MTBF}_I(\mathrm{t_i}) = \frac{\gamma(1+at_i^b)^2}{\theta(\mathrm{ab}t_i^{b-2}-(\mathrm{b}-1-at_i^b))}, \tag{15}$$

and the cumulative MTBF that can be found using Eq. (5) as follows:

$$\mathrm{MTBF}_c(\mathrm{t_i}) = \frac{\gamma \mathrm{t_i}(1+at_i^b)}{\theta\left(\mathrm{ab}t_i^{b-1}-\eta(1+at_i^b)\right)}. \tag{16}$$

3. Model Parameter Estimation

Parameter estimation is significantly important in the procedure of software reliability prediction. Non-Linear Least Squares Estimation (NLSE) method may be computationally simple but very effective in estimating model parameters. This technique is based on the observed failure data to determine the estimates of the model parameters, in this section we present the NLSE method for estimating the parameters of the modified NHPP LL model. Given the failure time data $(\mathrm{t_i}, \mathrm{y_i})$, where $\mathrm{y_i}$ is the cumulative number of faults detected by time $\mathrm{t_i}$ for $\mathrm{i} = (1, 2, \dots, \mathrm{n})$ and $0 < \mathrm{t_1} < \mathrm{t_2} < \mathrm{t_3} \dots < \mathrm{t_n}$ the NLSE method is to minimize the objective function defined by:

$$\mathcal{S}_{NLS}(\Theta) = \sum_{i=1}^{n}[\mathrm{y_i} - \mathrm{m}(\mathrm{t_i};\Theta)]^2, \quad (17)$$

where $\mathrm{m}(\mathrm{t_i};\Theta)$ is the mean value function at time $\mathrm{t_i}$, Θ is the unknown parameters of a NHPP model. The resulting estimates of parameters are obtained by minimizing Eq. (17). Traditionally, Gauss-Newton method or Lenvenberg-Marquardt algorithm is used to solve the optimization problems $\arg\min_{(\Theta)} \mathcal{S}_{NLS}(\Theta)$ [10]. Hence, the NLSE method of the modified NHPP LL model aims to minimize the following objective function:

$$\mathcal{S}_{NLS}(\theta, a, b) = \sum_{i=1}^{n}\left[\mathrm{y_i} - \frac{\theta\left(\mathrm{ab}t_i^{b-1} - \eta\left(1 + at_i^{b}\right)\right)}{\gamma(1 + at_i^{b})}\right]^2. \quad (18)$$

Taking the partial derivatives of Eq.(18) with respect to model parameters, and setting them equal to zero, we have:

$$\frac{\partial \mathcal{S}_{NLS}(\theta, a, b)}{\partial \theta} = 0 \Longrightarrow$$

$$\hat{\theta} = \sum_{i=1}^{n}\frac{\mathrm{y_i}\left(\mathrm{ab}t_i^{b-1} - \eta\left(1 + at_i^{b}\right)\right)}{\gamma\left(1 + at_i^{b}\right)} \Big/ \sum_{i=1}^{n}\left(\frac{\left(\mathrm{ab}t_i^{b-1} - \eta\left(1 + at_i^{b}\right)\right)}{\gamma(1 + at_i^{b})}\right)^2. \quad (19)$$

$$\frac{\partial \mathcal{S}_{NLS}(\Theta)}{\partial a} = 0 \Longrightarrow$$

$$2\sum_{i=1}^{n}\left(\mathrm{y_i} - \frac{\hat{\Theta}\left(\mathrm{ab}t_i^{b-1} - \eta\left(1 + at_i^{b}\right)\right)}{\gamma(1 + at_i^{b})}\right)\left(\frac{\hat{\Theta}\mathrm{b}t_i^{b-1}}{\gamma(1 + at_i^{b})^2}\right) = 0. \quad (20)$$

$$\frac{\partial \mathcal{S}_{NLS}(\Theta)}{\partial b} = 0 \Longrightarrow$$

$$2\sum_{i=1}^{n}\left(\mathrm{y_i} - \frac{\hat{\Theta}\left(\mathrm{ab}t_i^{b-1} - \eta\left(1 + at_i^{b}\right)\right)}{\gamma(1 + at_i^{b})}\right)\left(\frac{\hat{\Theta}\mathrm{a}t_i^{b-1}\left(1 + \mathrm{bl}\theta t_\mathrm{i} + at_i^{b}\right)}{\gamma\left(1 + at_i^{b}\right)^2}\right) = 0. \quad (21)$$

Solving Eqs. (20) and (21) numerically, we can obtain the point estimates of parameters *a* and *b*, then by substituting these estimates in Eq.(19) $\hat{\theta}$ can be obtained.

4. Goodness of Fit Criteria

Mean Square Error (MSE) calculates the variation between the predicted and actual values of observations. It is defined as [11]:

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n}\left(\mathrm{y_i} - \hat{\mathrm{m}}(\mathrm{t_i})\right)^2}{n-k}. \quad (22)$$

Smaller MSE indicates better fit model.

The coefficient of multiple determinations R^2 can be obtained as follows [11]:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{m}(t_i)\right)^2}{\sum_{i=1}^{n}\left(y_i - \sum_{k=1}^{n} y_k/n\right)^2}. \tag{23}$$

R^2 is a measure of the variation between the actual and fitted model. It takes the values in the range of 0 to 1, larger R^2 value represents better accuracy of the fitted model. Clearly, a value close to one of R^2 is highly appropriate. where y_i the total cumulated number of errors observed within time is $(0, t_i]$, $\hat{m}(t_i)$ is the estimated mean value function at time t_i, n is the number of errors in the software and k is the number of parameters in the model.

5. Data Analysis

In this section, we give some numerical examples using the following six published datasets to analyze the characteristics of learning effects:

(1) NTDS data: is obtained from [12], which created from the U.S. Navy Fleet Computer Programming Center, were observed during the software development phase for the real-time multicomputer complex system that is the central part of the Navel Tactical Data System (NTDS), the time between successive failures is (in days).
(2) S27 data: contains 41 time between successive failures (in seconds), reported by [13].
(3) DS3 data: is from [14] , represents 15 time between successive failures (in hours) of air conditioning equipment case 2.
(4) S2 data: were presented by [13], consists of 54 time between successive failures (in seconds).
(5) DS5 data: were used in [15], consists of 14 time between successive failures (in hours) for aircraft generator.
(6) DS6 data: is from [14], represents 23 time between successive failures (in hours) of air conditioning equipment case 1.

Tables [1-6] list the six failure data sets and Figure 1 represents them graphically.

Table 1: NTDS data.

9	12	11	4	7	2	5	8	5	7	1	6	1	9	4	1	3
3	6	1	11	33	7	91	2	1	87	47	12	9	135	258	16	35

Table 2: S27 data.

20336	11776	40933	34794	17136	148446	7995	1636	15830	21932
2485	11000	2880	61182	4800	38005	16200	6000	1000	10000
220	35580	81000	643095	47857	154800	170460	108540	73800	1860
336600	268140	74880	286200	25320	7080	59820	87900	76200	89280
1209600									

Table 3: DS3 data.

74	57	48	29	502	12	70	21	29	386	59	27	153	26	326

Table 4: S2 data.

191	222	280	290	290	385	570	610	365	390	275	360
800	1210	407	50	660	1507	625	912	638	293	1212	612
675	1215	2715	3551	800	3910	6900	3300	1510	195	1956	135
661	50	729	900	180	4225	15600	0	0	300	9021	2519
6890	3348	2750	6675	6945	7899						

Table 5: DS5 data.

577	742	403	662	902	136	147	79	164	579	39	111	45	10

Table 6: DS6 data.

413	14	58	37	100	65	9	169	447	184	36	201	118	34	31
18	18	67	57	62	7	22	34							

Table 7: Criteria results of comparison of the models.

Data set	Learning factors γ	η	Evaluation criteria MSE criteria	R^2
NTDS data n=34	0.3	0.7	3.5842	0.9650
	0.5	0.5	3.0213	0.9705
	0.7	0.3	2.7800	0.9728
S27 data n= 41	0.3	0.7	1.3924	0.9905
	0.5	0.5	1.418	0.9904
	0.7	0.3	1.505	0.9898
Ds3 data n= 15	0.3	0.7	0.6214	0.9712
	0.5	0.5	0.7397	0.9657
	0.7	0.3	0.6248	0.9710
S2 data n= 54	0.3	0.7	1.0238	0.9959
	0.5	0.5	1.1494	0.9954
	0.7	0.3	1.5050	0.9940
Ds5 data n=14	0.3	0.7	0.0742	0.9961
	0.5	0.5	0.0968	0.9949
	0.7	0.3	0.1404	0.9926
Ds6 data n=23	0.3	0.7	3.4798	0.9278
	0.5	0.5	3.5435	0.9265
	0.7	0.3	3.8471	0.9202

First, the estimation of the parameters for each model by using the NLSE method is performed. Then, the MSE and coefficient of determination R^2 criteria are evaluated, the results are summarized in Table 7.

According to this table, generally when the learning factor is greater than the autonomous errors-detected factor smaller MSE and higher R^2 are obtained and accordingly more efficient model.

In Figure 2, reliability considering the two factors for each model is showing a decrease graph, higher model's reliability is shown with the larger values of the learning factor and so more capable model.

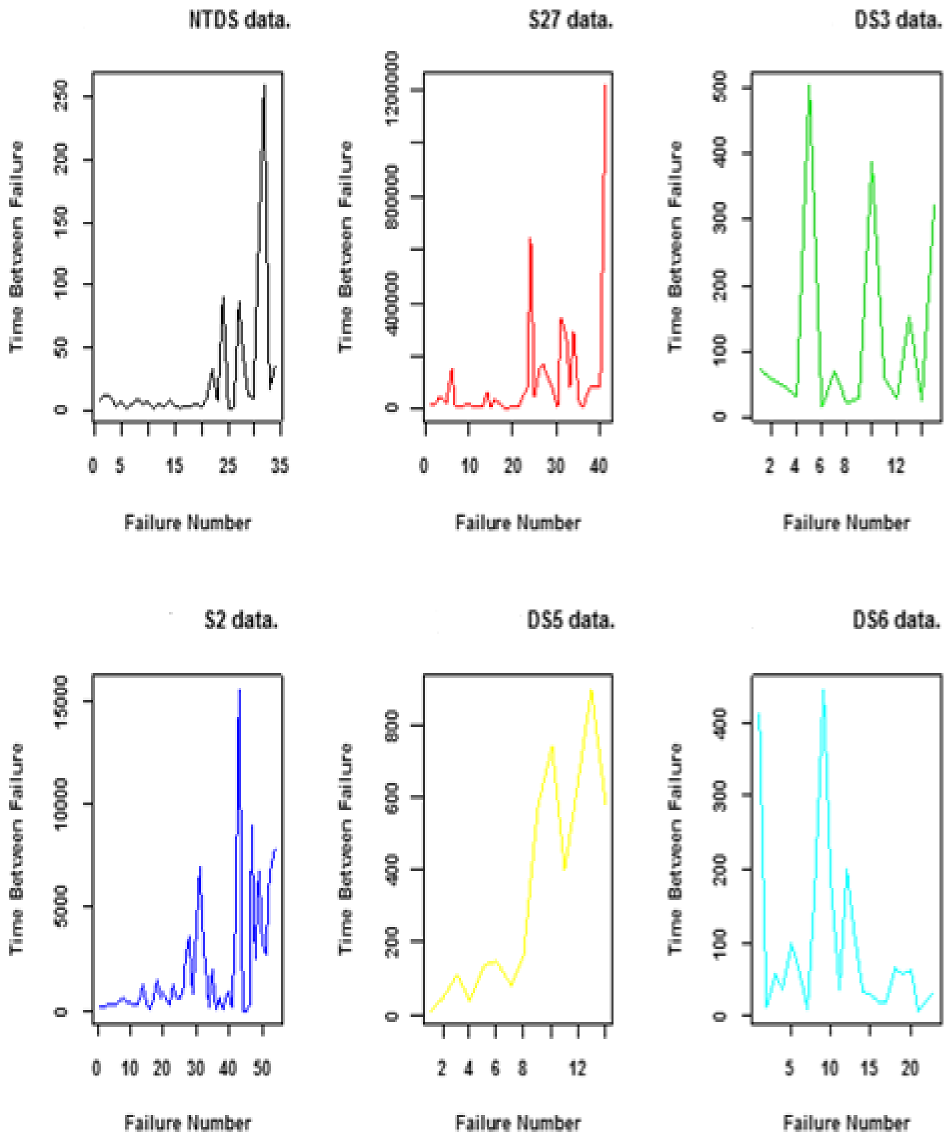

Figure 1: Time between failures versus failure number.

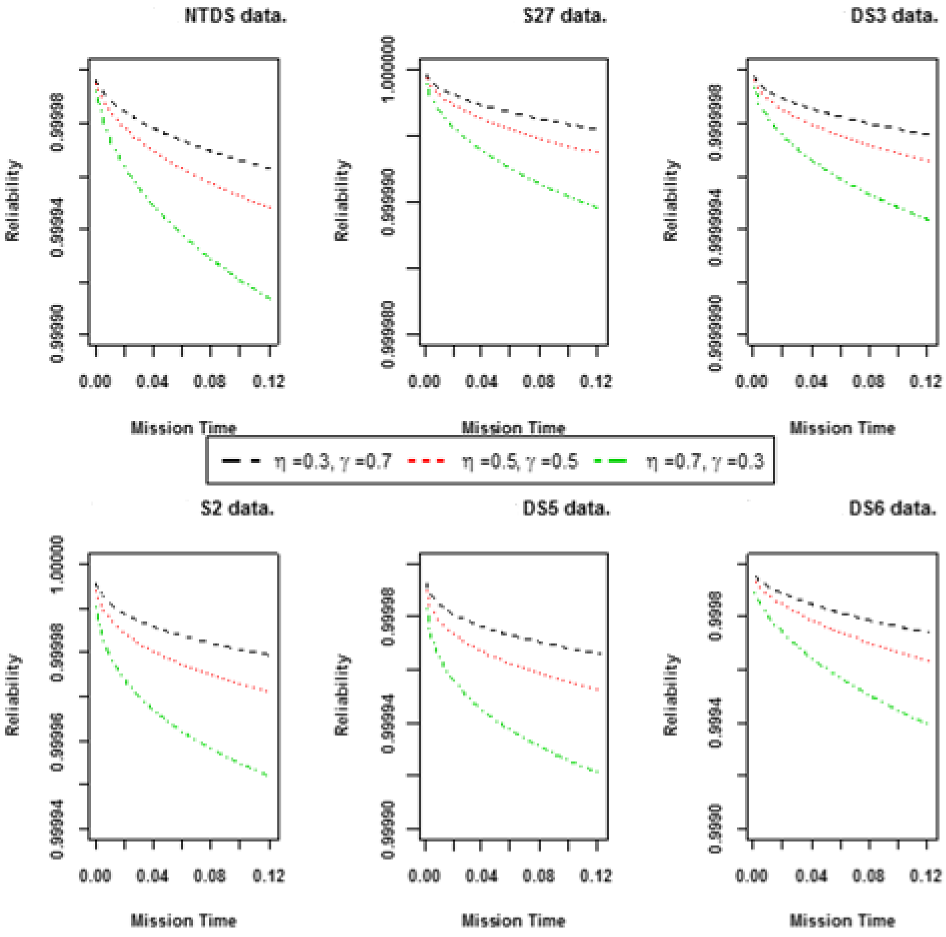

Figure 2: Reliability function of the models considering learning effects.

6. Conclusion

The reliability of a software system is usually accepted as the main issue in software quality since it measures software failures. During the testing phase of software development software quality is checked whether it meets the requirements or not. One way of monitoring testing phase to increase its accuracy is SRGM. SRGMs that incorporate factors that affect failure phenomena are more representative.

In this paper, a detailed analysis of the NHPP LL model incorporating learning effects is presented. Six real data sets have been used for the models' comparative analysis. According to our application the following points are concluded:

- When the value of the learning factor is the highest and the value of the autonomous errors-detected factor is the lowest, best fit model is obtained in terms of the MSE and coefficient of determination. Thus, higher values of the learning factor give more representative model.
- Regarding to the reliability figure, greater values of the learning factor has revealed higher reliability form.

References

[1] Yamada S. and Ohba H. (1983). “S-shaped software reliability modeling for software error detection”, IEEE Trans Reliab; 32:475–84.

[2] Ohba M. (1984). “Inflexion S-shaped software reliability growth models”, Stochastic Models in Reliability Theory (S. Osaki, Y. Hatoyama, Eds), Springer- Verlag Merlin, pp.144 – 162.

[3] Shyur H. J. (2003). “A stochastic software reliability model with imperfect debugging and change-point”, J Syst Software, 66:135–41.

[4] Ahmad N., Khan M.G.M., and Rafi L.S. (2011). “Analysis of an Inflection S-shaped Software Reliability Model Considering Log-logistic Testing-Effort and Imperfect Debugging”, International Journal of Computer Science and Network Security, Vol. 11 (1), pp. 161 – 171.

[5] Iqbal J., Ahmad N., and Quadri S. M. K. (2013). “A Software Reliability Growth Model with Two types of Learning”, Proceedings of the 1st IEEE International Conference on Machine Intelligence Research and Advancement, SMVDU, Jammu, India, pp. 498–503.

[6] Chiu K. (2011)."An improved model of software reliability growth under time-dependent learning effects", IEEE International Conference on Quality and Reliability, Bangkok, pp. 191-194. doi: 10.1109/ICQR.2011.6031707

[7] Iqbal, J. , Quadri , S.M.K. , and Ahmad, N. (2014). An Imperfect-Debugging Model with Learning-Factor Based Fault-Detection Rate. Proceedings of the 2014 IEEE International Conference on Computing for Sustainable Global Development (INDIA Com), (pp. 383-387).

[8] Lyu, M. R. (2002). Software Reliability Theory, Encyclopedia of Software Engineering, Wiley, pp. 1611-1630.

[9] Kuei-Chen, C., Yeu-Shiang, H., and Tzai-Zang, L. (2008). “A study of software reliability growth from the perspective of learning effects”, Reliability Engineering and System Safety 93, pp. 1410–1421.

[10] Marquardt, D. (1963). An algorithm for least-squares estimation of non-linear parameters. SIAM Journal of Applied Mathematics, 11(2), pp. 431-441.

[11] Singh V. B., Kapur P. K., and Mashaallah Basirzadeh. (2011). "Open Source Software Reliability Growth Model by Considering Change-Point." BIJIT - BVICAM’s International Journal of Information Technology, Vol. 4 No. 1, pp.405 – 410.

[12] Goel, A.L. and Okumoto, K. (1979). Time-Dependent Error-Detection Rate Model for Software Reliability and other Performance Measures, IEEE Trans. Reliability, R-28, 3, pp. 206-211.

[13] Lyu, M. R. (1996), Handbook of Software Reliability Engineering, IEEE Computer Society Press, McGraw-Hill, New York .

[14] Cox D.R., Lewis P. A.W. (1966). The Statistical Analysis of Series of Events. Methuen & Co., Ltd., London; John Wiley & Sons, Inc., New York.

[15] Black S.E., Rigdon S.E. (1996). Statistical inference for a modulated power law process. Journal of Quality Technology, 28.1:81-90.

7. Author Information

Lutfiah Ismail Al turk is currently working as associate professor of mathematical statistics in Statistics

Department at Faculty of Sciences, King AbdulAziz University, Jeddah, Kingdom of Saudi Arabia. Lutfiah Ismail Al turk obtained her B.Sc degree in statistics and computer science from Faculty of Sciences, King AbdulAziz University in 1993 and M.Sc (mathematical statistics) degree from Statistics Department, Faculty of Sciences, King AbdulAziz University in 1999. She received her Ph.D in mathematical statistics from university of Surrey, UK in 2007. Her current research interests include software reliability modeling and statistical machine learning.

Email: lturk@kau.edu.sa

URL: http://lturk.kau.edu.sa

Address: P.O. Box 42713 Jeddah 21551. Kingdom of Saudi Arabia.