American Journal of Theoretical and Applied Statistics
2016; 5(3): 80-86
http://www.sciencepublishinggroup.com/j/ajtas
doi: 10.11648/j.ajtas.20160503.11
ISSN: 2326-8999 (Print); ISSN: 2326-9006 (Online)

Regression Approach to Parameter Estimation of an Exponential Software Reliability Model

Albert Orwa Akuno[*], Timothy Mutunga Ndonye, Janiffer Mwende Nthiwa, Luke Akong'o Orawo

Department of Mathematics, Egerton University, Egerton, Kenya

Email address:

orwaakuno@gmail.com (A. O. Akuno), mutungatn@gmail.com (T. M. Ndonye), janienthiwa@yahoo.com (J. M. Nthiwa), orawo2000@yahoo.com (L. A. Orawo)

[*]Corresponding author

To cite this article:

Albert Orwa Akuno, Timothy Mutunga Ndonye, Janiffer Mwende Nthiwa, Luke Akong'o Orawo. Regression Approach to Parameter Estimation of an Exponential Software Reliability Model.*American Journal of Theoretical and Applied Statistics*. Vol. 5, No. 3, 2016, pp. 80-86. doi: 10.11648/j.ajtas.20160503.11

Received: March 10, 2016; **Accepted**: April 5, 2016; **Published**: April 21, 2016

Abstract: Mathematical studies about the likelihood of failures of software systems have been advanced by various researchers. These studies have modeled the behavior of software systems by using failure times and time between failures in the past. The Goel-Okumoto software reliability model is amongst the many software reliability models proposed to model the failure behavior of software systems. To be able to use the model in software reliability assessment, it is important to estimate its parameters *α and β* and the intensity function *λ(t)*. In this paper, classical parametric regression methods have been utilized in the estimation of the parameters *α and β*, the intensity function and the mean time between failures of the Goel-Okumoto software reliability model. The parameters *α and β* and the mean time between failures (MTBF) of the Goel-Okumoto software model have been estimated using the maximum likelihood estimation (MLE) method, regression approach applied to the model and simple linear regression model without assuming the Goel-Okumoto model. When these three estimation methods were validated using root mean squared error (RMSE) and mean absolute value difference (MAVD), which are the common error measurement criteria, regression approach applied to the Goel-Okumoto model outperformed MLE and simple linear regression estimation methods.

Keywords: Goel-Okumoto model, Regression Approach, Maximum Likelihood Estimation

1. Introduction

Various software reliability growth models have been proposed in the last three decades. The models enable software vendors to predict the behavior of software systems before a decision is made to release or to ship the software to users. Amongst the many software reliability growth models is the Goel – Okumoto software reliability model, a Non-Homogeneous Poisson process (NHPP) with intensity function

$$\lambda(t) = \alpha\beta e^{-\beta t} \qquad (1)$$

where α and β are parameters and t is the failure time. The software reliability model with intensity function given in Equation (1) was proposed by [1] in 1979 hence the name Goel-Okumoto (1979) software reliability model. The model is also called an exponential software reliability model. The reliability and the behavior of the software systems are studied by estimating the parameters of the software growth models. Various parameter estimation criteria have been advanced by different researchers in the past. These methods include but are not limited to, maximum likelihood estimation (MLE) method, least squares method, interval estimation and particle swam optimization method. Most researchers, for instance, [2], [3] and [4] have considered estimation of the parameters of Goel-Okumoto (1979) software reliability model whose intensity function is given in Equation (1) using MLE criteria. Literature from various research, for instance, [5, 6] and [7] have indicated that the Goel-Okumoto software reliability model is a good model to represent TBF of software systems.

In this work, based on the Goel-Okumoto software reliability growth model, predictive properties of mean time to failure (MTTF) and thus the estimators of the parameters α and β are computed using three methods; MLE method, regression approach using logarithm of the software failure data with Goel-Okumoto software reliability model assumption and simple linear regression applied directly to the software failuredata. The performance of the three methods of estimation is evaluated using RMSE and MAVD, which are the commonly used performance error measurement criteria in predictive analyses.

Reference [8] considered the point estimation of the power law process using regression approaches and the results were comparable to the traditional methods of estimation.

1.1. Methodology

What follows in this section is the methodology upon which this paper is based. We define mean time to failure (MTTF) and mean time between failures (MTBF) as is frequently used in reliability studies. We also provide software reliability data that will be used in illustrating the derived methods and procedures in section 2.

1.1.1. Mean Time to Failure

Mean time to failure (MTTF) is the average interval of time expected to the next failure time. In other words, given the reliability function $R(t)$, MTTF is a measure of the average time to failure for system with life distribution $F(t)$.

1.1.2. Mean Time Between Failure

The Mean Time Between Failures (MTBF) is the expected interval length from the current failure time, say, $T_n = t_n$ to the next failure time $T_{n+1} = t_{n+1}$. Let $f(t|t_1, t_2, ...t_n)$ denote the conditional distribution of failure time T_{n+1} given $T_1 = t_1, T_2 = t_2, ..., T_n = t_n$, then the MTBF is defined by

$$MTBF = \int_{t_n}^{\infty} f(t|t_1, t_2, ...t_n)\,dt - t_n$$

The reciprocal of the intensity function $\frac{1}{\lambda(t)}$ is used to represent the expected time to the next failure time, given that the n^{th} failure time occurred at time t, that is, $\frac{1}{\lambda(t)}$ is considered as the MTBF. Under special conditions, MTBF can be approximated by $\frac{1}{\lambda(t)}$. That is,

$$M\hat{T}BF = \frac{1}{\lambda(t)} \tag{2}$$

1.1.3. Mean Residual Time

Let T be a continuous random variable denoting failure time and in the interval $(0, \infty)$. The mean residual time (MRT) is the average time to the next failure given that no failure occurs up to time t and is defined by

$$m(t) = E(T - t|T > t)$$

The theorem under section 2.2.2 shows the relationship between MRT and reliability.

1.1.4. Software Failure Data

The following software failure data obtained from [4] has been used for the purposes of estimation and analysis in this study. The data is given in form of TBF, failure times (cumulative time between failure) and the failure number.

Table 1.*Time between failures data.*

Failure No.	Time between failures	Cumulative time between failures	Failure No.	Time between failures	Cumulative time between failures
1	30.02	30.02	16	15.53	151.78
2	1.44	31.46	17	25.72	177.50
3	22.47	53.93	18	2.79	180.29
4	1.36	55.29	19	1.92	182.21
5	3.43	58.72	20	4.13	186.34
6	13.2	71.92	21	70.47	256.81
7	5.15	77.07	22	17.07	273.88
8	3.83	80.90	23	3.99	277.83
9	21	101.90	24	176.06	453.93
10	12.97	114.87	25	81.07	535.00
11	0.47	115.34	26	2.27	537.27
12	6.23	121.57	27	15.63	552.90
13	3.39	124.96	28	120.78	673.68
14	9.11	134.07	29	30.81	704.49
15	2.18	136.25	30	34.19	738.68

Reference [7] argued that the software failure data given in Table 1 follow the Goel-Okumoto (1979) software reliability model.

1.2. Performance Error Measurement

In this section, we establish the metrics that will be used to evaluate the performance of the estimation models. There are various performance error measurement tools including but not limited to root mean squared error (RMSE) and mean absolute value difference (MAVD). Since we will use RMSE and MAVD in evaluating the performance of the three estimation models, it suffices to define them. These performance error measurement criteria are defined and explained in sections 1.2.1 and 1.2.2 respectively.

1.2.1. Root Mean Squared Error

Root mean squared error (RMSE) is the criteria most commonly used in error measurement, especially in prediction. The mean squared error (MSE) of an estimator $T = T(x_1, x_2, ..., x_n)$ of an observable parameter θ is defined by

$$MSE(T) = E\left[(T-\theta)^2\right]$$

Let TBF be the actual time between failures and $M\hat{T}BF$ be the predicted mean time between failures. The RMSE used in this paper is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(TBF_i - M\hat{T}BF\right)^2} \qquad (3)$$

1.2.2. Mean Absolute Value Difference

Mean absolute value difference (MAVD) is defined as the average of the absolute difference between predicted mean time between failures and actual times between failure values. The MAVD is defined as

$$MAVD = \frac{1}{n}\sum_{i=1}^{n}\left|TBF_i - M\hat{T}BF\right| \qquad (4)$$

2. Derivation of the Methods

In this section, we derive the three methods of estimation of the Goel-Okumoto software reliability parameters and its MTBF. In section 2.1, we consider the MLE method while the regression model and the resulting intensity function is derived in section 2.2. Finally, simple linear regression model and the resulting intensity function is considered in section 2.3.

2.1. Maximum Likelihood Estimation

The joint probability distribution function of the failure times $T_1, T_2, ..., T_n$ from a Non-Homogeneous Poisson process with intensity function $\lambda(t)$ is given as; [9]

$$f(t_1, t_2, ..., t_n) = \left(\prod_{i=1}^{n}\lambda(t_i)\right)\exp\left(-\int_0^T \lambda(t)\,dt\right) \qquad (5)$$

Under the assumption that the failure times follow the Goel-Okumoto software reliability model with intensity function as in Equation (1), the joint probability distribution function of the failure times is given as

$$f(t_1, t_2, ..., t_n) = \alpha^n \beta^n e^{-\beta\sum_{i=1}^{n} t_i} e^{-\alpha(1-e-\beta t_n)} \qquad (6)$$

Taking the log-likelihood function of Equation (6) gives

$$L = n\ln\alpha + n\ln\beta - \beta\sum_{i=1}^{n} t_i - \alpha\left(1-e^{-\beta t_n}\right) \qquad (7)$$

Differentiating L partially with respect to α and β and equating to zero gives

$$\frac{\partial L}{\partial \alpha} = \frac{n}{\alpha} - \left(1-e^{-\beta t_n}\right) = 0 \qquad (8)$$

$$\frac{\partial L}{\partial \beta} = \frac{n}{\beta} - \sum_{i=1}^{n} t_i - \alpha t_n e^{-\beta t_n} = 0 \qquad (9)$$

Solving Equations (8) and (9) for α and β, we obtain the ML estimators denoted by $\hat{\alpha}_{mle}$ and $\hat{\beta}_{mle}$ as

$$\hat{\alpha}_{mle} = \frac{n}{1-e^{-\hat{\beta}_{mle}t_n}} \qquad (10)$$

$$\frac{n}{\hat{\beta}_{mle}} = \sum_{i=1}^{n} t_i + \frac{n t_n e^{-\hat{\beta}_{mle}t_n}}{1-e^{-\hat{\beta}_{mle}t_n}} \qquad (11)$$

It has been shown [10] that the necessary and sufficient condition for Equations (10) and (11) to have a unique and positive solutions $\hat{\alpha}_{mle}$ and $\hat{\beta}_{mle}$ is if and only if $\left(2\sum_{i=1}^{n} t_i \Big/ n\right) < t_n$.

A numerical procedure known as the Newton Raphson method can be used to iteratively solve Equations (10) and (11) and use the MLEs thus obtained to obtain an estimator of the MTBF as

$$M\hat{T}BF = \left(\hat{\alpha}_{mle}\hat{\beta}_{mle}\right)^{-1} e^{\hat{\beta}_{mle}t_n} \qquad (12)$$

We denote the model from Equation (12) as $\mathrm{MTBF}_a - \mathrm{mle}$ model.

2.2. Regression Model and the Resulting Intensity Function.

Subsection 2.2.1 outlines the derivation of the regression model and the resulting intensity function is derived in subsection 2.2.2.

2.2.1. Regression Approach for the Goel-Okumoto Software Reliability Model

This study stems from the fact that the logarithm of the intensity function of the Goel-Okumoto software reliability model is a linear function of the software failure times. It is thus proposed that the model can be taken as a simple linear regression. The parameters of the model are estimated using the classical regression approaches. References [11], [12], [13] and [14] used the inverse of the power law process, which is a NHPP to approximate MTBF. Since the Goel-Okumoto software reliability model is also a NHPP, its MTBF can be approximated by taking the inverse of its intensity function as

$$MTBF = \frac{1}{\lambda(t)} = \frac{1}{\alpha\beta e^{-\beta t}} \tag{13}$$

where t is the failure time.

Taking natural logarithm both sides of Equation (13) we get

$$\ln(MTBF) = -\ln(\alpha\beta) + \beta t \tag{14}$$

Let

$$Y = \ln(MTBF) \tag{15}$$

$$-\ln(\alpha\beta) = a \tag{16}$$

$$\beta = b \tag{17}$$

Then Equation (14) becomes

$$Y = a + bt \tag{18}$$

Using the method of least squares for the linear regression model, the least squares estimators of the parameters a and b in Equation (18) are obtained as

$$\hat{b} = \frac{\sum_{i=1}^{n} t_i Y_i - \frac{\left(\sum_{i=1}^{n} t_i\right)\left(\sum_{i=1}^{n} Y_i\right)}{n}}{\sum_{i=1}^{n} t_i^2 - \frac{\left(\sum_{i=1}^{n} t_i\right)^2}{n}} \tag{19}$$

and

$$\hat{a} = \overline{Y} - \hat{b}\overline{t} \tag{20}$$

After obtaining the estimators of a and b as in Equations (19) and (20) we get the estimators of the Goel-Okumoto software model parameters α and β denoted by $\hat{\alpha}_{reg}$ and $\hat{\beta}_{reg}$ from Equations (16) and (17) as;

$$\hat{\alpha}_{reg} = \frac{e^{-\hat{a}}}{\hat{b}} \tag{21}$$

and

$$\hat{\beta}_{reg} = \hat{b} \tag{22}$$

The estimator of MTBF can be obtained from Equation (2) and the regression estimators in Equations (9) and (10) as;

$$M\hat{T}BF = \left(\hat{\alpha}_{reg}\hat{\beta}_{reg}\right)^{-1} e^{\hat{\beta}_{reg} t_n} \tag{23}$$

We call this $\text{MTBF}_b - \text{regression}$ model.

2.2.2. Intensity Function for the Regression Model

In order to derive the resulting intensity function from the assumed linear relationship in Equation (18), we state the following theorem without proof.

Theorem

Let T be a random variable of continuous type with density function $f(t)$ and the cumulative density function $F(t)$. If it is assumed that $f(t) = F(t) = 0$ for $t < 0$, then

$$E(T) = \int_0^\infty (1 - F(t))dt = \int_0^\infty R(t)dt \tag{24}$$

and the MRT is given as

$$M(t) = \frac{\int_t^\infty R(u)du}{R(t)} \tag{25}$$

From the assumed linear relationship $Y = \ln(MTTF) = a + bt$, we get

$$MTTF = e^{a+bt} = e^a e^{bt} \tag{26}$$

Equating the MRT in the above theorem and the MTTF, i.e. equating Equations (25) and (26) in order to obtain the intensity failure function $\lambda(t)$, we have $\frac{\int_t^\infty R(u)du}{R(t)} = e^a e^{bt}$ from which we obtain

$$e^a e^{bt} R(t) = \int_t^\infty R(u)du \tag{27}$$

By differentiating Equation (27) and using the result

$$\frac{d}{dt}\left\{\int_t^\infty R(u)du\right\} = -\frac{d}{dt}\left\{\int_\infty^t R(u)du\right\} = -R(t)$$

We obtain

$$be^{a}e^{bt}R(t)+e^{a}e^{bt}\frac{d}{dt}R(t)=-R(t) \quad (28)$$

If we let $e^{a}=k$, the Equation (28) becomes

$$kbe^{bt}R(t)+ke^{bt}\frac{d}{dt}R(t)=-R(t) \quad (29)$$

Re-arranging Equation (29), we obtain

$$\frac{d}{dt}R(t)+\left(\frac{1}{ke^{bt}}+\frac{kbe^{bt}}{ke^{bt}}\right)R(t)=0 \quad (30)$$

It is known that

$$\lambda(t)=\frac{f(t)}{R(t)} \quad (31)$$

But

$$f(t)=\frac{dF(t)}{dt}=\frac{d(1-R(t))}{dt}=-\frac{dR(t)}{dt} \quad (32)$$

From Equation (32), Equation (31) becomes

$$\lambda(t)=\left.-\frac{dR(t)}{dt}\middle/R(t)\right. \quad (33)$$

Now, from Equation (30), we have $\frac{-d}{dt}R(t)=\left(\frac{1+kbe^{bt}}{ke^{bt}}\right)R(t)$. Thus Equation (33) becomes

$$\lambda(t)=\frac{1+kbe^{bt}}{ke^{bt}} \quad (34)$$

Equation (34) is the intensity function obtained when we assume a linear regression equation from the Goel-Okumoto software reliability model.

2.3. Simple Linear Regression Model and the Resulting Intensity Function

Subsection 2.3.1 outlines the derivation of the simple linear regression model and the derivation of resulting intensity function from the simple linear regression model is outlined in subsection 2.3.2.

2.3.1. Simple Linear Regression Model

In this section, we directly take a simple linear regression model instead of assuming the Goel-Okumoto (1979) reliability model. That is, we assume that the failure times and TBF are linearly related as

$$\mathrm{TBF}=b+at+\varepsilon \quad (35)$$

where TBF is the dependent variable and time of failure t is the independent variable and a and b are constants that need to be estimated. ε represents the error term.

Using least squares method, the estimators of the parameters a and b in Equation (35) are obtained as

$$\hat{a}=\frac{\sum_{i=1}^{n}t_i(TBF_i)-\frac{\left(\sum_{i=1}^{n}t_i\right)\left(\sum_{i=1}^{n}(TBF_i)\right)}{n}}{\sum_{i=1}^{n}t_i^2-\frac{\left(\sum_{i=1}^{n}t_i\right)^2}{n}} \quad (36)$$

and

$$\hat{b}=\overline{\mathrm{TBF}}-\hat{a}\overline{t} \quad (37)$$

where $\overline{\mathrm{TBF}}$ denotes the average time between software failure. Thus the prediction equation (38) represents the estimating mean time between software failures.

$$M\hat{T}BF=\hat{b}+\hat{a}t \quad (38)$$

We denote the estimator of MTBF from the simple linear regression model as $\mathrm{MTBF_c}$ – Linear regression model.

2.3.2. Intensity Function for the Simple Linear Regression Model

Here, we derive the intensity function resulting from the simple linear regression model in Equation (35) using the MRT. For a simple linear regression Equation (35),

$$MTTF=at+b \quad (39)$$

Equating the MRT in Equation (13) and MTTF in Equation (39) we have $\frac{\int_t^{\infty}R(u)du}{R(t)}=at+b$ from which we obtain

$$(at+b)R(t)=\int_t^{\infty}R(u)du \quad (40)$$

Differentiating Equation (40) and using the procedures and steps in Section 2.2.2, it can easily be shown that the intensity function resulting from the assumption of the simple linear regression model in Equation (35) is;

$$\lambda(t)=\frac{1+a}{at+b} \quad (41)$$

3. Results and Comparison of the Performance of the Three Estimation Methods

This section is divided into two where the results obtained from the three methods of estimation are discussed in section 3.1 and thereafter, the performances of these three methods are compared using RMSE and MAVD in section 3.2.

3.1. Results from the Three Methods of Estimation

The results obtained from the MLE method, regression method and simple linear regression method are respectively given in subsections 3.1.1, 3.1.2 and 3.1.3.

3.1.1. Using Maximum Likelihood Estimation Method

From Equations (10) and (11) and the data in Table 1, the MLE of the parameters α and β of the Goel-Okumoto software reliability model with intensity function given in Equation (1) are $\hat{\alpha}_{mle} = 31.698171$ and $\hat{\beta}_{mle} = 0.003962$. Using these estimates and Equation (12), we find MSE and MAVD of the failure data in Table 1 as is in Table 2.

Table 2. *MSE and MAVD of the* $\mathrm{MTBF_a - mle}$ *model.*

Model	RMSE	MAVD
$\mathrm{MTBF_a - mle}$	679.1673	125.8273

3.1.2. Using Regression Model

Using Equation (21) and Equation (22), we find the estimates of the parameters α and β of the Goel-Okumoto software reliability as $\hat{\alpha}_{reg} = 70.3433$ and $\hat{\beta}_{reg} = 0.003426915$

Using these estimates, we find MSE and MAVD of the failure data in Table 1 as is in Table 3.

Table 3*. MSE and MAVD of the* $\mathrm{MTBF_b - regression}$ *model.*

Model	RMSE	MAVD
$\mathrm{MTBF_b - regression}$	150.7669	43.118

3.1.3. Using Simple Linear Regression

Using Equations (37) and (38), we obtain the simple linear regression estimates of the parameters α and β of the Goel-Okumoto software reliability model as $\hat{\alpha} = 2.260013$ and $\hat{\beta} = 0.09329616$. The following is the MSE and MAVD for the data in Table 1 obtained using simple linear regression approach based on Equation (38).

Table 4. *MSE and MAVD for* $\mathrm{MTBF_c - Linear\ regression}$ *model.*

Model	RMSE	MAVD
$\mathrm{MTBF_c - Linear\ regression}$	254.9832	57.5122

3.2. Comparison of the Three Methods of Estimation

Comparison of the performance of the three methods of estimation of the parameters α and β and MTBF of the Goel-Okumoto software reliability model based on RMSE and MAVD is given in Table 5.

Table 5*. RMSE and MAVD for* $\mathrm{MTBF_a - mle}$ *,* $\mathrm{MTBF_b - regression}$ *and* $\mathrm{MTBF_c - Linear\ regression}$ *models.*

Model	RMSE	MAVD
$\mathrm{MTBF_a - mle}$	679.1673	125.8273
$\mathrm{MTBF_b - regression}$	150.7669	43.118
$\mathrm{MTBF_c - Linear\ regression}$	254.9832	57.5122

Based on the results from Table 5, $\mathrm{MTBF_b - regression}$ is the best model for estimating the parameters α and β and MTBF for the Goel-Okumoto (1979) software reliability model. This is so because the model has the least RMSE and MAVD. It is worth noting that this method of estimation performs better than pure linear regression model and MLE method and thus should be preferred. Based on this model and the data in Table 1, the preferred estimates of α and β and MTBF are thus obtained as $\hat{\alpha}_{reg} = 70.3433$ and $\hat{\beta}_{reg} = 0.003426915$ $\mathrm{M\hat{T}BF} = 52.14881$ respectively.

4. Conclusions

Estimation of the parameters of software reliability models using the traditional techniques like the maximum likelihood method and the least squares Method pose some difficulties since the models are generally in non-linear relationships, [15]. The derivation and calculation of the MLEs usually require specialized software and more powerful computers for solving the non-linear equations. Some researchers, for instance, [16] argue that the difficulty experienced in the computations of MLE is less of a problem as time goes by as more statistical packages are being developed to contain and solve the complex maximum likelihood (ML) equations. However, these statistical packages require more complex algorithms and programming languages for them to work. MLEs are also heavily biased when there is small data on failure times, [17]. In this paper, we have presented a simpler and more efficient parameter estimation method for the Goel – Okumoto software reliability model. This stems from the fact that the logarithm of the intensity function of the model is a linear function of the software failure times and the parameters can thus be estimated using the traditional least squares regression method. The estimates thus obtained are better than MLE which is the widely used method in estimating the parameters of the model. It is also worth noting that when the parameters of the model are estimated using simple linear regression method, the results obtained are still better than MLE method.

References

[1] Goel, A. L. and Okumoto, K., (1979). Time-dependent error detection rate model for software reliability and other performance measures. *IEEE Trans. Reliability*, 28: 206–211.

[2] Stringfellow, C. and Amschler, A. A., (2002). An Empirical Method for Selecting Software Reliability Growth Models. *Empirical Software Engineering*, 7: 319-343.

[3] Meyfroyt, P. H. A., (2012). *Parameter Estimation for Software Reliability Models*. Masters Thesis, Eindhoven University of Technology, Eindhoven, Netherlands.

[4] Xie, M., Goh, T. N and Ranjan, P., (2002). *Some effective control chart procedures for reliability monitoring*. Elsevier, Reliability engineering and System safety.

[5] Akuno, A. O., Orawo, L. A. andIslam, A. S. (2014) One-Sample Bayesian Predictive Analyses for an Exponential Non-Homogeneous Poisson Process in Software Reliability. *Open Journal of Statistics*, 4, 402-411.

[6] Akuno, A. O., Orawo, L. A. and Islam, A. S. (2014) Two-Sample Bayesian Predictive Analyses for an Exponential Non-Homogeneous Poisson Process in Software Reliability. *Open Journal of Statistics*, 4, 742-750.

[7] Satya, P., Bandla, S. R. and Kantham, R. R. L., (2011). Assessing Software Reliability using Inter Failures Time Data. *International Journal of Computer Applications*, 18: 975-978.

[8] Abdelah, M. M., (2006*). Regression Approach to Software Reliability Models*. Graduate Theses and Dissertations, University of South Florida, USA.

[9] Rigdon, S. E. and Basu, A. P., (2000). The Power Law Process: a Model for the Reliability of Repairable systems. *Journal of Quality Technology*, 21: 251-260.

[10] Hossain, S. A. and Dahiya, R. C., (1993). Estimating the Parameters of a Non- homogenous Poisson-Process Model for Software Reliability. *IEEE Tranis- actions on Reliability*, 42:604-612.

[11] Ascher, H. and Feingold, H., (1984). Repairable Systems Reliability, Inference, Misconceptions and their Causes. Marcel Dekker, New York.

[12] Cox, D. R. and Lewis, P. A., (1996). *The Statistical Analysis of Series of Events*. Chapman and Hall, London.

[13] Roberts, H., (2000). *Predicting the Performance of Software Systems via the Power Law Process*. Ph.D. thesis, University of South Florida, Tampa, FL.

[14] Suresh, N., (1992). *Modeling and Analysis of Software Reliability*. Ph.D. thesis, University of South Florida, Tampa, FL.

[15] Karambir, B. and Adima. A., (2014). A review on Parameter Estimation Techniques of Software Reliability Growth Models. *International Journal of Computer Applications Technology and Research*, 4: 267-272, ISSN: 2319-8656.

[16] Latha, S. and Lilly, F., (2012). A Comparison of Parameter Best Estimation Method for Software Reliability Models. *International Journal of Software Engineering & Applications (IJSEA)*, Vol. 3, No. 5.

[17] Xie, M., Hong, G. Y. and Wohlin, C., (1997). A Practical Method of the Estimation of Software Reliability Growth in the Early Stages of Testing. *Proceedings IEEE 7th International Symposium on Software Reliability Engineering*. pp. 116-123, Albuquerque, USA.