ISSN:0975-9646

Detection of Burr type XII Reliable Software Using SPRT on Interval Domain Data

Dr. R. Satya Prasad[1], B. Ramadevi[2], Dr. G.Sridevi[3]

[1]*Assoc.Prof, Department of CSE, Acharya Nagarjuna University, Guntur, India.*
[2]*Research Scholor, Department of CSE, Acharya Nagarjuna University, Guntur, India.*
[3] *Professor, Dept. of CSE, KL University, Vaddeswaram, Guntur, India.*

Abstract– **As the volumes of data/software is getting increased in the internet day by day, there is a need for the people to have the tools/mechanism to assess the software reliability as it takes more time to come to conclusion. In Classical Hypothesis first of all testing volumes of data is to be collected and later the conclusions are to be drawn which may take more time. In this paper a well known test procedure of statistical science called as Sequential Probability Ratio Test (SPRT) is adopted for Burr Type XII model in assessing the reliability of developed software. It requires considerably less number of observations when compared with the other existing testing procedures. Hence Sequential Analysis of Statistical Science could be adopted to decide upon the reliable / unreliable of the developed software very quickly. Besides the present paper proposes the performance of SPRT on Interval domain data using Burr type XII model and analyzed the results by applying on 6 data sets. The Maximum Likelihood Estimation is used for estimation of parameters.**

Keywords**: **Burr **Type XII model, Sequential Probability Ratio Test, MLE, Software Reliability, NHPP**.

I. INTRODUCTION

The SPRT was initially developed by Wald (1947) for quality control problems during World War II. It has many extensions and applications: such as in clinical trial and in quality control. The original development of the SPRT is used as a statistical device to decide which of two simple hypotheses is more correct. Wald's SPRT is currently the only Bayesian Statistical procedure in SISA. What is required in Bayesian statistics is quite a detailed description of the expectations of the outcome under the model prior to executing the data collection. In Wald's SPRT, if certain conditions are met during the data collection decisions are taken with regard to continuing the data collection and the interpretation of the gathered data. Wald's procedure is particularly relevant if the data is collected sequentially. Sequential Analysis is different from Classical Hypothesis Testing were the number of cases tested or collected is fixed at the beginning of the experiment. In Classical Hypothesis Testing the data collection is executed without analysis and Consideration of the data. After all data is collected the analysis is done and conclusions are drawn. However, in Sequential Analysis every case is analyzed directly after being collected, the data collected up to that moment is then compared with certain threshold values, incorporating the new information obtained from the freshly collected case. This approach allows one to draw conclusions during the data collection, and a final conclusion can possibly be reached at a much earlier stage as is the case in Classical Hypothesis Testing. The advantages of Sequential Analysis are easy to see. As data collection can be terminated after fewer cases and decisions taken earlier, the savings in terms of human life and misery, and financial savings, might be considerable. In the analysis of software failure data we often deal with either Time between Failures or failure count in a given time interval. If it is further assumed that the average number of recorded failures in a given time interval is directly proportional to the length of the interval and the random number of failure occurrences in the interval is explained by a Poisson process then we know that the probability equation of the stochastic process representing the failure occurrences is given by a homogeneous Poisson process with the expression

$$P[N(t)=n]=\frac{e^{-\lambda t}(\lambda t)^n}{n!} \tag{1.1}$$

(Stieber 1997) observes that if classical testing strategies are used, the application of software reliability growth models may be difficult and reliability predictions can be misleading. However, he observes that statistical methods can be successfully applied to the failure data. He demonstrated his observation by applying the well-known sequential probability ratio test of (Wald 1947) for a software failure data to detect unreliable software components and compare the reliability of different software versions. In this paper we consider popular SRGM Burr Type XII model and adopt the principle of Stieber in detecting unreliable software components in order to accept or reject the developed software. The theory proposed by Stieber is presented in Section 2 for a ready reference. Extension of this theory to the SRGM – Burr Type XII is presented in Section 3.Maximum Likelihood parameter estimation method is presented in Section 4. Application of the decision rule to detect unreliable software components with respect to the proposed SRGM is given in Section 5.

II. SEQUENTIAL TEST FOR A POISSON PROCESS

The sequential probability ratio test was developed by A. Wald at Columbia University in 1943. Due to its

Usefulness in development work on military and naval equipment it was classified as ‘Restricted’ by the Espionage Act (Wald 1947). A big advantage of sequential tests is that they require fewer observations (time) on the average than fixed sample size tests. SPRTs are widely used for statistical quality control in manufacturing processes. An SPRT for homogeneous Poisson processes is described below. Let $\{N(t),t \geq 0\}$ be a homogeneous Poisson process with rate ‘λ’. In our case, N(t) = number of failures up to time ‘t’ and ‘λ’ is the failure rate (failures per unit time). Suppose that we put a system on test (for example a software system, where testing is done according to a usage profile and no faults are corrected) and that we want to estimate its failure rate ‘λ’. We cannot expect to estimate ‘λ’ precisely. But we want to reject the system with a high probability if our data suggest that the failure rate is larger than λ_1 and accept it with a high probability, if it’s smaller than λ_0. As always with statistical tests, there is some risk to get the wrong answers. So we have to specify two (small) numbers ‘α’ and ‘β’, where ‘α’ is the probability of falsely rejecting the system. That is rejecting the system even if $\lambda \leq \lambda_0$. This is the "producer’s" risk. β is the probability of falsely accepting the system .That is accepting the system even if $\lambda \geq \lambda_1$. This is the “consumer’s” risk. With specified choices of λ_0 and λ_1 such that $0 < \lambda_0 < \lambda_1$, the probability of finding N(t) failures in the time span (0,t) with λ_1,λ_0 as the failure rates are respectively given by

$$P_1 = \frac{e^{-\lambda_1 t}[\lambda_1 t]^{N(t)}}{N(t)!} \qquad (2.1)$$

$$P_0 = \frac{e^{-\lambda_0 t}[\lambda_0 t]^{N(t)}}{N(t)!} \qquad (2.2)$$

The ratio $\frac{P_1}{P_0}$ at any time ‘t’ is considered as a measure of deciding the truth towards λ_0 *or* λ_1, given a sequence of time instants say $t_1 < t_2 < \cdots < t_k$ and the corresponding realizations $N(t_1), N(t_2) \ldots N(t_k)$ *of* $N(t)$.Simplification of $\frac{P_1}{P_0}$ gives

$$\frac{P_1}{P_0} = exp(\lambda_0 - \lambda_1)\, t + \left[\frac{\lambda_1}{\lambda_0}\right]^{N(t)}$$

The decision rule of SPRT is to decide in favour of λ_1, *in favor of* λ_0 or to continue by observing the number of failures at a later time than 't' according as $\frac{P_1}{P_0}$ is greater than or equal to a constant say A, less than or equal to a constant say B or in between the constants A and B. That is, we decide the given software product as unreliable, reliable or continue (Satya Prasad 2007) the test process with one more observation in failure data, according as

$$\frac{P_1}{P_0} \geq A \qquad (2.3)$$

$$\frac{P_1}{P_0} \leq B \qquad (2.4)$$

$$B < \frac{P_1}{P_0} < A \qquad (2.5)$$

The approximate values of the constants A and B are taken as $A \cong \frac{1-\beta}{\alpha}$, $B \cong \frac{\beta}{1-\alpha}$

Where ‘α ’ and ‘β ’ are the risk probabilities as defined earlier. A simplified version of the above decision processes is to reject the system as unreliable if N(t) falls for the first time above the line

$$N_U(t) = a.t + b2 \qquad (2.6)$$

To accept the system to be reliable if N(t) falls for the first time below the line

$$N_L(t) = a\,t - b_1 \qquad (2.7)$$

To continue the test with one more observation on $[t, N(t)]$ as the random graph of $[t, N(t)]$ is between the two linear boundaries given by Eq. (2.6) and (2.7) where

$$a = \frac{\lambda_1 - \lambda_0}{\log\left[\frac{\lambda_1}{\lambda_0}\right]} \qquad (2.8)$$

$$b_1 = \frac{log\left[\frac{1-\alpha}{\beta}\right]}{log\left[\frac{\lambda_1}{\lambda_0}\right]} \qquad (2.9)$$

$$b_2 = \frac{log\left[\frac{1-\beta}{\alpha}\right]}{log\left[\frac{\lambda_1}{\lambda_0}\right]} \qquad (2.10)$$

The parameters α, β, λ_0 *and* λ_1 can be chosen in several ways. One way suggested by Stieber (1997) is

$$\lambda_0 = \frac{\lambda \log(q)}{q-1}$$

$$\lambda_1 = q\,\frac{\lambda \log q}{q-1} \quad where\ q = \frac{\lambda_1}{\lambda_0}$$

If λ_0 *and* λ_1 are chosen in this way, the slope of $N_U(t)$*and* $N_L(t)$ equals λ. The other two ways of choosing λ_0 *and* λ_1 are from past projects (for a comparison of the projects) and from part of the data to compare the reliability of different functional areas(components).

III SEQUENTIAL TEST FOR SOFTWARE RELIABILITY GROWTH MODELS

In Section 2, for the Poisson process we know that the expected value of $N(t) = \lambda(t)$ called the average number of failures experienced in time 't' .This is also called the mean value function of the Poisson process. On the other hand if we consider a Poisson process with a general function (not necessarily linear) $m(t)$ as its mean value function the probability equation of such a process is

$$P[N(t) = Y] = \frac{[m(t)]^y}{y!}\, e^{-m(t)}, y = 0,1,2 \ldots$$

Depending on the forms of $m(t)$ we get various Poisson processes called NHPP, for our Burr type XII model. The mean value function is given as

$$m(t) = a\left[1 - \left(1 + t^c\right)^{-b}\right], \quad t \geq 0$$

We may write

$$P_1 = \frac{e^{-m_1 t}[m_1 t]^{N(t)}}{N(t)!}$$

$$P_0 = \frac{e^{-m_0 t}[m_0 t]^{N(t)}}{N(t)!}$$

Where $m_1(t), m_0(t)$ are values of the mean value function at specified sets of its parameters indicating reliable software and unreliable software respectively. The mean value function $m(t)$ contains the parameters $'a','b'and\ 'c'$. Let P_0, P_1 be values of the NHPP at two specifications of b say $b_0, b_1\ where\ (b_0 < b_1)\ and$ two specifications of c say $c_0, c_1\ where\ (c_0 < c_1)$. It can be shown that for our model $m(t)$ at b_1 is greater than that at b_0 and $m(t)$ *at* c_1 is greater than that at c_0. Symbolically $m_0(t) < m_1(t)$. . Then the SPRT procedure is as follows:

Accept the system to be Reliable if $\frac{P_1}{P_0} \leq B$

$$i.e., \quad \frac{e^{-m_1(t)}[m_1(t)]^{N(t)}}{e^{-m_0(t)}[m_0(t)]^{N(t)}} \leq B$$

$$i.e.,\ N(t) \leq \frac{log\left(\frac{\beta}{1-\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - log m_0(t)} \quad (3.1)$$

Decide the system to be unreliable and Reject if $\frac{P_1}{P_0} \geq A$

$$i.e., \quad \frac{e^{-m_1(t)}[m_1(t)]^{N(t)}}{e^{-m_0(t)}[m_0(t)]^{N(t)}} \geq A$$

$$i.e.,\ N(t) \geq \frac{log\left(\frac{\beta}{1-\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - log m_0(t)} \quad (3.2)$$

Continue the test procedure as long as

$$\frac{log\left(\frac{\beta}{1-\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - log m_0(t)} < N(t) < \frac{log\left(\frac{1-\beta}{\alpha}\right) + m_1(t) - m_0(t)}{\log m_1(t) - log m_0(t)} \quad (3.3)$$

Substituting the appropriate expressions of the respective mean value function $m(t)$, we get the respective decision rules and are given in followings lines.

Acceptance Region:

$$N(t) \leq \frac{\log\left(\frac{\beta}{(1-\alpha)}\right) + a\left[\left(1+t^{c_0}\right)^{-b_0} - \left(1+t^{c_1}\right)^{-b_1}\right]}{\log a\left[\frac{\left(1+t^{c_0}\right)^{-b_0}}{\left(1+t^{c_1}\right)^{-b_1}}\right]} \quad (3.4)$$

Rejection Region:

$$N(t) \geq \frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left[\left(1+t^{c_0}\right)^{-b_0} - \left(1+t^{c_1}\right)^{-b_1}\right]}{\log a\left[\frac{\left(1+t^{c_0}\right)^{-b_0}}{\left(1+t^{c_1}\right)^{-b_1}}\right]} \quad (3.5)$$

Continuation Region:

$$\frac{\log\left(\frac{\beta}{(1-\alpha)}\right) + a\left[\left(1+t^{c_0}\right)^{-b_0} - \left(1+t^{c_1}\right)^{-b_1}\right]}{\log a\left[\frac{\left(1+t^{c_0}\right)^{-b_0}}{\left(1+t^{c_1}\right)^{-b_1}}\right]}$$

$$< N(t) <$$

$$\frac{\log\left(\frac{1-\beta}{\alpha}\right) + a\left[\left(1+t^{c_0}\right)^{-b_0} - \left(1+t^{c_1}\right)^{-b_1}\right]}{\log a\left[\frac{\left(1+t^{c_0}\right)^{-b_0}}{\left(1+t^{c_1}\right)^{-b_1}}\right]} \quad (3.6)$$

It may be noted that in the proposed model the decision rules are exclusively based on the strength of the sequential procedure (α, β) and the values of the respective mean value functions namely, $m_0(t)$, $m_1(t)$. If the mean value function is linear in $'t'$ passing through origin, that is, $m(t) = \lambda t$ the decision rules become decision lines as described by (Stieber 1997). In that sense equations (3.1), (3.2) , (3.3) can be regarded as generalizations to the decision procedure of Stieber(1997). The applications of these results for live software failure data are presented with analysis in Section 5.

IV Maximum Likelihood Estimation

In this section we develop expressions to estimate the parameters of the Burr type XII model based on interval domain data. Parameter estimation is of primary importance in software reliability prediction.

A set of failure data is usually collected in one of two common ways, time domain data and interval domain data. In this paper parameters are estimated from the interval domain data.

The mean value function of Burr type XII model is given by

$$m(t) = a\left[1 - \left(1+t^c\right)^{-b}\right] \quad (4.1)$$

In order to have an assessment of the software reliability, a, b and c are to be known or they are to be estimated from software failure data. Expressions are now delivered for estimating 'a', 'b' and 'c' for the Burr type XII model.

Assuming the given data are given for the cumulative number of detected errors n_i in a given time interval $(0, t_i)$ where i=1,2, ….. n and $0 < t_1 < t_2 < \ldots t_n$, then the logarithmic likelihood function (LLF) for interval domain data is given by

$$LogL=\sum_{i=1}^{k}(n_i-n_{i-1})\log\left[m(t_i)-m(t_{i-1})\right]-m(t_k) \tag{4.2}$$

$$LogL=\sum_{i=1}^{k}(n_i-n_{i-1})\log\left\{a\left[1-\left(1+t_i^c\right)^{-b}\right]-a\left[1-\left(1+t_{i-1}^c\right)^{-b}\right]\right\}-a\left[1-\left(1+t_k^c\right)^{-b}\right]$$

$$LogL=\sum_{i=1}^{k}(n_i-n_{i-1})\left\{Loga+\log\left[\left(1+t_{i-1}^c\right)^{-b}-(1+t_i^c)^{-b}\right]\right\}-a+a(1+t_k^c)^{-b} \tag{4.3}$$

Taking the Partial derivative with respect to ‘a’ and equating to ‘0’.

(i.e., $\frac{\partial Log\,L}{\partial a}=0$)

$$\therefore a=\sum_{i=1}^{k}(n_i-n_{i-1})\frac{(1+t_k^c)^b}{(1+t_k^c)^b-1} \tag{4.4}$$

The parameter ‘b’ is estimated by iterative Newton Raphson Method using

$b_{n+1}=b_n-\frac{g(b)}{g'(b)}$, Where $g(b) and\ g'(b)$ are expressed as follows.

$$g(b)=\frac{\partial LogL}{\partial b}=0$$

$$\frac{\partial Log\,L}{\partial b}=g(b)=\sum_{i=1}^{k}(\mathrm{n}_i-n_{i-1})\left\{\begin{array}{l}\left[-\log\left(t_{i-1}+1\right)-\log(t_i+1)+\frac{(t_i+1)^b\log(t_i+1)-(t_{i-1}+1)^b\log(t_{i-1}+1)}{(t_i+1)^b-(t_{i-1}+1)^b}\right]\\+\left[\frac{1}{(t_k+1)^b-1}Log\left(\frac{1}{1+t_k}\right)\right]\end{array}\right\} \tag{4.5}$$

Again partial differentiating with respect to ‘b’ and equate to 0 , we get

$$g'(b)=\frac{\partial^2 LogL}{\partial b^2}=0$$

$$\frac{\partial^2 LogL}{\partial b^2}=g'(b)=\sum_{i=1}^{k}(n_i-n_{i-1})\left[\frac{2(t_{i-1}+1)^b(t_i+1)^b\log(t_i+1)\log\left(\frac{t_{i-1}+1}{t_i+1}\right)}{\left[(t_i+1)^b-(t_{i-1}+1)^b\right]^2}\right]+\sum_{i=1}^{k}(n_i-n_{i-1})\log(1+t_k)\frac{(t_k+1)^b\log(t_k+1)}{\left[(t_k+1)^b-1\right]^2} \tag{4.6}$$

The parameter ‘c’ is estimated by iterative Newton Raphson Method using

$$c_{n+1}=c_n-\frac{g(c_n)}{g'(c_n)}$$

Where $g(c)\ and\ g'(c)$ are expressed as follows.

$$g(c)=\frac{\partial LogL}{\partial c}=0$$

$$\frac{\partial LogL}{\partial c} = g(c) = \sum_{i=1}^{k}(n_i - n_{i-1})\left[-\log t_{i-1}\frac{t_{i-1}^c}{(1+t_{i-1}^c)} - \log t_i\frac{t_i^c}{(1+t_i^c)} + \frac{t_i^c \log t_i - t_{i-1}^c \log t_{i-1}}{(\mathrm{t}_i^c - t_{i-1}^c)}\right] - \sum_{i=1}^{k}(\mathrm{n}_i - n_{i-1})\frac{\log t_k}{(1+t_k^c)}$$

(4.7)

$$g'(c) = \frac{\partial^2 LogL}{\partial c^2} = 0$$

$$\frac{\partial^2 Log\,L}{\partial c^2} = g^{'}(c) = \sum_{i=1}^{k}(n_i - n_{i-1})\begin{bmatrix}\left(Log\left(\frac{t_{i-1}}{t_i}\right)\frac{t_i^c.t_{i-1}^c}{(t_i^c - t_{i-1}^c)^2}\{\log t_i - \log t_{i-1}\}\right) \\ -(\log t_{i-1})^2.\frac{t_{i-1}^c}{(1+t_{i-1}^c)^2} - (\log t_i)^2\frac{t_i^c}{(1+t_{i-1}^c)^2}\end{bmatrix} + \sum_{i=1}^{k}(n_i - n_{i-1})(\log t_k)^2.\frac{t_k^c}{(1+t_k^c)^2}$$

(4.8)

TABLE 1. ESTIMATES OF a, b, c AND SPECIFICATIONS OF b_0, b_1, c_0, c_1

Data Set	Estimate of 'a'	Estimate of 'b'	b₀	b₁	Estimate of 'c'	c₀	c₁
Phase 1	25.994042	0.978993	0.478993	1.478993	1.083116	0.583116	1.583116
Phase 2	41.590454	0.978993	0.478993	1.478993	1.083119	0.583119	1.583119
Release 1	87.533224	0.978352	0.478352	1.478352	1.082376	0.582376	1.582376
Release 2	111.778470	0.977674	0.477674	1.477674	1.081290	0.581290	1.581290
Release 3	59.376054	0.971698	0.471698	1.471698	1.051525	0.551525	1.551525
Release 4	42.831021	0.977671	0.477674	1.477674	1.081287	0.581287	1.581287

TABLE 2. SPRT ANALYSIS FOR 6 DATA SETS

Data Set	T	N(t)	R.H.S. of equation (3.4) Acceptance region $(\le)$	R.H.S. of equation (3.5) Rejection region $(\ge)$	Decision
Phase 1	1	1	1.192028	1.856685	Accepted
Phase 2	1	3	1.792173	2.373032	Rejected
Release 1	1	16	3.338247	3.822450	Rejected
Release 2	1	13	4.089652	4.588755	Rejected
Release 3	1	6	2.430323	2.960544	Rejected
Release 4	1	1	1.839248	2.415563	Accepted

V SPRT ANALYSIS OF LIVE DATASETS

We see that the developed SPRT methodology is for a software failure data which is of the form [t, N(t)] where N(t) is the failure number of software system or its sub system in 't' units of time. In this section we evaluate the decision rules based on the considered mean value function for Six different data sets of the above form, borrowed from (Pham 2005), (Wood 1996). Based on the estimates of the parameter 'b' in each mean value function, we have chosen the specifications of $b_0 = b - \delta$, $b_1 = b + \delta$ $c_0 = c - \delta$ and, $c_1 = c + \delta$ equidistant on either side of b obtained through a data set to apply SPRT such that $b_0 < b < b_1$ and $c_0 < c < c_1$. Assuming the value of δ = 0.5, the choices are given in the Table 1

Using the selected b_0, b_1 *and* c_0, c_1 and subsequently the $m_0(t)$*and* $m_1(t)$ for each model we calculated the decision rules given by Equations (3.4), (3.5) sequentially at each 't' of the data set taking the strength (α, β) as (0.05, 0.2).These are presented for the model in Table 2.

From Table 2 we see that a decision either to accept or reject the system is reached much in advance of the last time instant of the data (the testing time).

VI. Conclusion

In this paper, SPRT procedure is applied on the proposed model to detect reliable/unreliable software products. The Table 2 shows that Burr type XII SRGM as exemplified for 6 datasets indicates that the model is performing well in arriving at a decision. This model has given a decision of Acceptance for 2 datasets, Rejection for 4 datasets. The result of the present study indicates that the model is performing well in arriving at a decision. Therefore, we may conclude that the model Burr type XII is most appropriate model to decide upon reliability / unreliability of software.

References

[1] GOEL, A.L and OKUMOTO, K. "*A Time Dependent Error Detection Rate Model For Software Reliability And Other Performance Measures*", IEEE Transactions on Reliability, vol.R-28, pp.206-211, 1979.

[2] Pham. H., "System software reliability", Springer. 2006.

[3] Satya Prasad, R., "Half logistic Software reliability growth model ", 2007, Ph.D Thesis of ANU,India.

[4] Wald. A., "Sequential Analysis", John Wiley and Son, Inc, New York. 1947.

[5] STIEBER, H.A. "*Statistical Quality Control: How To Detect Unreliable Software Components"*, Proceedings the 8th International Symposium on
Software Reliability Engineering, 8-12. 1997.

[6] WOOD, A. "*Predicting Software Reliability"*, IEEE Computer, 2253-2264. 1996.

[7] Xie, M., Goh. T.N., Ranjan.P., "Some effective control chart procedures for reliability monitoring" -Reliability engineering and System Safety 77 143 – 150¸ 2002.

[8] Michael. R. Lyu, "The hand book of software reliability engineering", McGrawHill & IEEE Computer Society press.

[9] Dr R.Satya Prasad, G.Krishna Mohan and Prof R R L Kantham. Article: Time Domain based Software Process Control using Weibull Mean Value Function. *International Journal of Computer Applications* 18(3):18-21, March 2011.

[10] Musa, J.D., Iannino, A., Okumoto, k., 1987. "Software Reliability: Measurement Prediction Application". McGraw – Hill, New York.

[11] Hee-cheul Kim., "Assessing Software Reliability based on NHPP using SPC", *International Journal of Software Engineering and its Applications*, vol.7,No.6 (2013), pp.61-70.