Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 66 (2014) 20514-20522

ISSN: 2229-712X

Identifying the reliable software using SPRT: An ordered statistics approach

V.Goutham[1,*] and R.Satya Prasad[2]

[1]Department of Computer Science and Engineering, St Mary College of Engineering Hyderabad, India.

[2]Department of Computer Science and Engineering, Acharya Nagarjuna University Guntur, India.

$Min \sum_{i=1}^{n} w_i . x_i$

$s.t\ A_{(n+w)\times n} X_{n\times 1} \ge b_{(n+w)\times 1}$

$Min \sum_{i=1}^{n} w_i . x_i$

$s.t\ y = A_{n\times n} X_{n\times 1} \ge b_{n\times 1}$

ARTICLE INFO

Article history:

Received: 17 November 2013;

Received in revised form:

22 December 2013;

Accepted: 1 January 2014;

Keywords

Reliable software,

Software quality,

Ordered Statistics,

Exponential Distribution,

NHPP,

SPRT, MLE.

ABSTRACT

Today software quality is an consequential parameter and to produce high quality reliable software and to keep a specific time schedule is a big challenge. There are many software reliability models that are based on the times of occurrences of errors in the debugging of software. To overcome the challenge many methodology and practices of software engineering have been evolved for developing reliable software. Improving algorithms of directing the manner of software development are underway. Order statistics are used in a broad category of exercised assignments their application in improvisation problems, determination of outliers, linear estimation. In this paper we proposed the detecting the exponential reliable software using SPRT on time domain data based on Ordered Statistics approach. The unknown parameters can be estimated using the Maximum Likelihood Estimation (MLE).

Introduction

Goel (1985) has defined software reliability as the probability that during a pre-specified testing or operational time interval, software faults do not cause a program to fail:"Let *F* be a class of faults, defined arbitrarily, and *T* be measure of relevant time, the units of which are dictated by the application at hand. Then the reliability of the software with respect to the class of faults *F* and with respect to the metric T, is the probability that no fault of the class occurs during the execution of the program for a pre-specified period of relevant time." Several classes of models have been proposed to capture this definition of reliability; among the most prominent are models built on the assumptions that waiting times between software failures are exponentially distributed [1]. The sequential probability ratio test (SPRT) is a specific sequential hypothesis test, developed by Abraham Wald. Neyman and Pearson's 1933 result inspired Wald to reformulate it as a sequential analysis problem.

Software reliability is the most important and most measurable aspect of software quality and it is very customer oriented. The user will also benefit from software reliability measure, because the user is concerned with efficient operation of the system. If the operational needs with respect to quality are in accurately specified, the user will either get a system at an excessively high price or with an excessively high operational cost. In Classical Hypothesis Testing, the data collection is executed without analysis and consideration of the data. After all the data is collected the analysis is done, conclusions are drawn where as sequential analysis is a method of statistical inference whose characteristic features is that number of observation required by the procedure is not determined in advance of the experiment. The decision to terminate the experiment depends, at each stage, on the results of the observation previously made. A merit of sequential method, as applied to testing statically hypothesis, is that test procedure can be constructed which require, on the average ,a substantially smaller number of observation that equally reliable test procedure based on a predetermined number of observations [3].

This paper describes a method for detecting software faults based on the Sequential Probability Ratio Test (SPRT) with 4^{th} and 5^{th} ordered statistics. The SPRT is the optimal statistical test that makes the correct decision in the shortest time among all tests that are subject to the same level of decision errors. As a result, it can be expected that the proposed method has the potential of providing the quickest detection of a fault compared with other methods with the same false-alarm and miss-detection rates. SPRT is issued to detect the fault based on the calculated likelihoods of the hypotheses. In the analysis of software failure data we often deal with number of recorded failures in a given time domain interval. If it is further assumed that the average number of recorded failures in a

Tele:

E-mail addresses: v.goutham@gmail.com

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 non homogeneous Poisson process with the expression.

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

Stieber (1997) observes that if classical testing strategies are used (no usage testing), 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 (SPRT) 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 a popular SRGM – proposed by Goel and Okumoto(1979) and adopt the principle of Stieber (1997) in detecting unreliable software components in order to accept/reject a developed software. The theory proposed "Order Statistics and Inference" by Balakrishnan. and Clifford Cohen is presented in Section 2 for a ready reference. Extension of this theory to the ordered statistics is described as model description .The procedure for parameter estimation is presented in Section 3 and the test process of wald's sequential test is presented in section 4 Application of the decision rule to detect unreliable software components with respect to the proposed SPRT is given in Section 5 (see [4][5][6]).The final conclusion is presented in Section 6[2]. The probabilistic models are applied to estimate software reliability with the field data. Various NHPP software reliability models are available to estimate the software reliability.

Ordered Statistics

Order statistics are used in a wide variety of practical situations. Order statistics deals with properties and applications of ordered random variables and of functions of these variables. The use of order statistics is significant when failures are frequent or inter failure time is less. Let X denote a continuous random variable with Probability Density Function (PDF) f(x) and Cumulative Distribution Function (CDF) F(x), and let (X1 , X2 , …, Xn) denote a random sample of size n drawn on X. The original sample observations may be unordered with respect to magnitude. A transformation is required to produce a corresponding ordered sample. Let (X1 , X2 , …, Xn) denote the ordered random sample such that X1 < X2 < … < Xn; then (X1, X2, …, Xn) are collectively known as the order statistics derived from the parent X. The various distributional characteristics can be known from Balakrishnan and Cohen [7]. The time-domain data represent the time lapse between every two consecutive failures. On the other hand if a reasonable waiting time for failures is not a serious problem, we can group the time-domain data into non overlapping successive sub groups of size 4 or 5 and add the failure times within each sub group. For instance if a data of 100 time-domain are available we can group them into 20 disjoint subgroups of size 5. The sum total in each subgroup would denote the time lapse between every 5th order statistics in a sample of size 5. In general for time-domain data of size "n", if r (any natural no) less than "n" and preferably a factor n, we can conveniently divide the data into "k" disjoint subgroups (k=n/r) and the cumulative total in each subgroup indicate the time between every rth failure. The probability distribution of such a time lapse would be that of the rth ordered statistics in a subgroup of size r, which would be equal to rth power of the distribution function of the original variable m (t).The whole process involves the mathematical model of the mean value function and knowledge about its parameters. If the parameters are known they can be taken as they are for the further analysis, if the parameters are not know they have to be estimated using a sample data by any admissible, efficient method of distribution. If software failures are quite frequent keeping track of time domain failure is tedious. If failures are more frequent order statistics are preferable[8]. The t_4 values are obtained by sum of each non overlapping four of the above failures in its sequential order [2]. That is, if X_1, X_2, ……, X_{60} are the inter failure times, then $\boldsymbol{Z_1 = \sum_{i=1}^{4} X_i}$, $\boldsymbol{Z_2 = \sum_{i=5}^{8} X_i}$, ……,$\boldsymbol{Z_{15} = \sum_{i=57}^{60} X_i}$ are the times to 4^{th} failure of size 15. Similarly t_5 values are obtained by sum of each non overlapping five of the above failures in its sequential order. That is, if X_1, X_2, … , X_{60} are the inter failure times, then $\boldsymbol{Z_1 = \sum_{i=1}^{5} X_i}$, $\boldsymbol{Z_2 = \sum_{i=6}^{10} X_i}$, ……,$\boldsymbol{Z_{10} = \sum_{i=56}^{60} X_i}$ .are the times of 5^{th} failures of size 10.

Model Description

Considering failure detection as a non homogenous Poisson process with an exponentially decaying rate function, the expected number of failures observed by time t is given by $m(t) = a(1 - e^{-bt})$ and the failure rate by $\lambda(t) = m'(t)$. To calculate the parameter values using Order Statistics approach, we considered exponential distribution (Juran, J. M. (ed.), 1988).

The mean value function of exponential distribution is

$$m(t) = a(1 - e^{-bt})$$

In order to group the time-domain data into non overlapping successive sub groups of size r the mean value function can be written as

$$m(t) = a(1 - e^{-bt})^r$$

$$m(s_k) = [a(1 - e^{-bs_k})]^r$$

$$m'(s_k) = a^r r\,(1 - e^{-bs_k})^{r-1}\, b e^{-bs_k} \quad \text{--- 3.1}$$

The Log likelihood function L can be written as

$$L = e^{-m(s_n)} \prod_{k=1}^{n} m'(s_k) \quad \text{----------3.2}$$

Substituting eq-3.1 in eq-3.2 we can write

$$L = e^{-m(s_n)} \prod_{k=1}^{n} a^r r\,(1 - e^{-bs_k})^{r-1}\, b e^{-bs_k}$$

$$\log L = -m(s_n) + \sum_{k=1}^{n} [\log a^r + \log b + \log r + \log e^{-bs_k} + \log(1 - e^{-bs_k})^{r-1}] \quad \text{-------------------------3.3}$$

$$m(s_n) = [a(1 - e^{-bs_n})]^r \quad \text{------------------------------3.4}$$

Substitute equation 3.4 in 3.3 we get

$$\log L = -[a(1 - e^{-bs_n})]^r + \sum_{k=1}^{n} [\log a^r + \log b + \log r + \log e^{-bs_k} + \log(1 - e^{-bs_k})^{r-1}] \quad \text{---------- 3.5}$$

$$\frac{\partial \log L}{\partial a} = 0,$$

$$a^r = \frac{n}{(1 - e^{-bs_n})]^r} \quad \text{------------- 3.6}$$

$$\frac{\partial \log L}{\partial b} = 0,$$

$$a^r\, r\, s_n e^{-bs_n} (1 - e^{-bs_n})]^{r-1} + \frac{n}{b} + 0 - \sum_{k=1}^{n} s_k + (r-1) \sum_{k=1}^{n} \frac{s_k e^{-bs_k}}{(1 - e^{-bs_k}} \quad \text{----------------- 3.7}$$

Substitute equation 3.6 in 3.7 we obtain the following equation

$$g(b) = \frac{nr\, s_n e^{-bs_n}}{(1 - e^{-bs_n})} + \frac{n}{b} - \sum_{k=1}^{n} s_k + (r-1) \sum_{k=1}^{n} \frac{s_k e^{-bs_k}}{(1 - e^{-bs_k}} \quad \text{-----------3.8}$$

Derivate with respect to b of equation 3.8 we obtain

$$g'(b) = \frac{-nrs_n^2 e^{-bs_n}}{(1 - e^{-bs_n})^2} - \frac{n}{b^2} - (r-1) \sum_{k=1}^{n} \frac{(s_k^2 e^{-bs_k}}{(1 - e^{-bs_k})^2} \quad \text{------ 3.9}$$

Parameter Estimation

Parameter estimation is a statistical method trying to estimate parameters based on inter failures time data which is based on ordered statistics. For the given observations using equations 3.8 and 3.9 the parameters '$\hat{a}$' and '$\hat{b}$ are computed by using the popular Newton Rapson method (Pham. H., 2003) .Based on the time between failures data given in Table-6.1 , we compute the parameter through ordered statistics. We use cumulative time between failures data for software reliability through SPRT. The parameters obtained from exponential distribution applied on the given time domain data are as follows:

Table 3.1.1: Parameter estimates of 4 and 5 order Statistics

Data Set	Order	$\hat{a}$	$\hat{b}$
SYS2 Data	4	2.14497	0.000064
	5	1.764372	0.000069

'$\hat{a}$' and '$\hat{b}$' are the parameters of ordered statistics and the values can be computed using analytical method for the given time between failures data shown in Table 3.1.1.

Wald's Sequential Probability Ratio Test

In this section Wald's sequential probability ratio test, SPRT, and some of its basic properties are described. A good introduction to the theory of sequential analysis can be found, for example, in [9][10][11] . The sequential probability ratio test was developed by A.Wald at Columbia University in 1943.In statistical terms, software system fault detection is essentially a binary decision or hypothesis testing problem, it is either a fault or no fault. Statistical theory and methods for such a problem are fairly complete and mature. Statistical tests are available that are simple and optimal under various criteria.

All methods for hypothesis test may be sequential or nonsequential. In a nonsequential method, a fixed size of samples (i.e., a fixed number of measurements) is used and the decision is made based on this whole block of samples altogether. A sequential method uses the samples one by one and the decision may be made at any time when sufficient evidence is gathered. A sequential test consists of a stopping rule, which determines when the test is done, and a decision rule, determines which hypothesis to choose [9].

Let {N(t),t ≥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 (0 < λ 0 < λ 1) . 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 λ ≤λ0. This is the "producer's" risk. β is the probability of falsely accepting the system .That is accepting the system even if λ ≥ λ 1. This is the "consumer's" risk. With specified choices of λ 0 and λ 1 such that 0 < λ 0 < λ 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

$$P1 = \frac{[\lambda 1 t]^{N(t)} . e^{-\lambda t}}{N(t)!} \quad \text{--------------(4.1)}$$

$$P0 = \frac{[\lambda 0 t]^{N(t)} . e^{-\lambda t}}{N(t)!} \quad \text{--------------(4.2)}$$

The ratio $\frac{P1}{P0}$ at any time 't' is considered as a measure of deciding the truth towards λ0 or λ1, given a sequence of time instants say t1<t2<t3 and the corresponding realizations $N(t1), N(t2), \ldots\ldots N(tK)$ of N(t). Simplification of $\frac{P1}{P0}$ gives

$$\frac{P1}{P0} = \exp(\lambda 0 - \lambda 1)\, t + \left[\frac{\lambda 1}{\lambda 0}\right]^{N(t)}$$

The decision rule of SPRT is to decide in favor of λ1 , in favor of λ0 or to continue by observing the number of failures at a later time than 't' according as P1/P0 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 the test process with one more observation in failure data, according as

$$\frac{P1}{P0} \geq A \quad \text{--- (4.3)}$$

$$\frac{P1}{P0} \geq B \quad \text{--- (4.4)}$$

$$B < \frac{P1}{P0} < A \quad \text{-------------------------------------- (4.5)}$$

The approximate values of the constants A and B are taken as

$$A \approx \frac{1-\beta}{\alpha}, \qquad B \approx \frac{\beta}{1-\alpha}$$

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

$$N_U(t) = a.t + b_2 \quad \text{---(4.6)}$$

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

$N_L(t) = a.t - b_1$ --(4.7)

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 equations (4.6) and (4.7) where

a = $\frac{\lambda 1-\lambda 0}{\log(\frac{\lambda 1}{\lambda 0})}$ ---(4.8)

b1 = $\frac{\log[(1-\alpha)/\beta]}{\log[\frac{\lambda 1}{\lambda 0}]}$ --(4.9)

b2 = $\frac{\log[(1-\beta)/\alpha]}{\log[\frac{\lambda 1}{\lambda 0}]}$ --(4.10)

The parameters α, β , λ 0 and λ1 can be chosen in several ways. One way suggested by Stieber (1997) is

λ0 = $\frac{\lambda.\log(q)}{q-1}$, λ1= q. $\frac{\lambda.\log(q)}{q-1}$

Where q = $\frac{\lambda 1}{\lambda 0}$

If λ0 and λ1 are chosen in this way, the slope of N_U (t) and N_L (t) equals λ. The other two ways of choosing λ0 and λ1 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).

Sequential Test for Software reliability Growth Models

In Section 4, for the Poisson process we know that the expected value of N(t) = λt 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 a such a process is

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

Depending on the forms of m(t) we get various Poisson processes called NHPP[11]. For our model the mean value function is given as $m(t)=a(1-e^{-bt})$ where a>0,b>0,t>0 We may write P1 = $\frac{[m1(t)]^{N(t)}.e^{-m1(t)}}{N(t)!}$

P0 = $\frac{[m0(t)]^{N(t)}.e^{-m0(t)}}{N(t)!}$

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. For instance the model we have been considering its m(t) function, contains a pair of parameters a,b with ' a' as a multiplier. Also a, b are positive. Let P_0 , P_1 be values of the NHPP at two specifications of b say b_0 , b_1 where (b0< *b*1 *)* respectively. It can be shown that for our models m(t) at b_1 is greater than that at b_0. Symbolically $m_0(t)<m_1(t)$.Then the SPRT procedure is as follows:

Accept the system to be reliable $\frac{P1}{P0} \leq$ B

ie: $\frac{e^{-m1(t)} \; [m1(t)]^{N(t)}}{e^{-m0(t)} \; [m0(t)]^{N(t)}} \leq$ B

ie: N(t) ≤ $\frac{\log(\frac{\beta}{1-\alpha})+m1(t)-m0(t)}{\log m1(t)-\log m0(t)}$ ----------------------------------- (5.1)

Decide the system to be unreliable and reject if $\frac{P1}{P0} \geq$ A

ie: N(t) ≥ $\frac{\log(\frac{1-\beta}{\alpha})+m1(t)-m0(t)}{\log m1(t)-\log m0(t)}$ -------------------------------------- (5.2)

Continue the test procedure as long as

$$\frac{\log\left(\frac{1-\beta}{\alpha}\right)+m1(t)-m0(t)}{\log m1(t)-\log m0(t)} \leq N(t) \leq \frac{\log\left(\frac{\beta}{1-\alpha}\right)+m1(t)-m0(t)}{\log m1(t)-\log m0(t)} \quad \text{---(5.3)}$$

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

$m(t)=a(1-e^{-bt})$

Acceptance region:

$$N(t) \leq \frac{\log\left(\frac{\beta}{1-\alpha}\right)+m1(t)-m0(t)}{\log m1(t)-\log m0(t)} \quad \text{where } m(t)=a(1-e^{-bt})$$

$$N(t) \leq \frac{\log\left(\frac{\beta}{1-\alpha}\right)+a(e^{-b0t}-e^{-b1t})}{\log\left[\frac{1-e^{-b1t}}{1-e^{-b0t}}\right]} \quad \text{------------------------------------(5.4)}$$

Rejection region:

$$N(t) \geq \frac{\log\left(\frac{1-\beta}{\alpha}\right)+a(e^{-b0t}-e^{-b1t})}{\log\left[\frac{1-e^{-b1t}}{1-e^{-b0t}}\right]} \quad \text{------------------------------------(5.5)}$$

Continuation region:

$$\frac{\log\left(\frac{\beta}{1-\alpha}\right)+a(e^{-b0t}-e^{-b1t})}{\log\left[\frac{1-e^{-b1t}}{1-e^{-b0t}}\right]} < N(t) < \frac{\log\left(\frac{1-\beta}{\alpha}\right)+a(e^{-b0t}-e^{-b1t})}{\log\left[\frac{1-e^{-b1t}}{1-e^{-b0t}}\right]} \quad \text{----------(5.6)}$$

It may be noted that in the above 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) = λt decision rules become decision lines as described by Stieber (1997).

In that case equations (5.1), (5.2) , (5.3) 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 6. The equations of lines only depends on α ,β and $\frac{\lambda 1}{\lambda 0}$ (see [4] and [5][12]).

Numerical Illustration

The procedure of a for failure software process will be illustrated with an example here. Table 6.1 show the time between failures of software product represented as per the size of the data sets. Based on the estimates of the parameter ‘b’ in each mean value function, we have chosen the specifications of b0=b-δ, b1=b+δ, equidistant on either side of estimate of b obtained through a data set to apply SPRT such that b0 < b < b1. Assuming the value of δ= 0.000025, the choices are given in the following table. Using the selected b0 and b1 subsequently the m_0(t),m_1(t) for the model.

We calculated the decision rules given by Equations 5.4, 5.5, sequentially at each‘t’ of the data sets taking the strength (α,β) as (0.05, 0.2). These are presented for the model in Table 4.

Table 6.1: SYS2 Data (Michael R. Lyu, 1996a)

Failure .No	*Time Between Failures (hrs)*	*Failure .No*	*Time Between Failures (hrs)*	*Failure .No*	*Time Between Failures (hrs)*	*Failure .No*	*Time Between Failures (hrs)*
1	479	23	437	45	460	67	1866
2	266	24	340	46	565	68	490
3	277	25	405	47	1119	69	1487
4	554	26	535	48	437	70	4322
5	1034	27	277	49	927	71	1418
6	249	28	363	50	4462	72	1023
7	693	29	522	51	714	73	5490
8	597	30	613	52	181	74	1520
9	117	31	277	53	1485	75	3281
10	170	32	1300	54	757	76	2716
11	117	33	821	55	3154	77	2175
12	1274	34	213	56	2115	78	3505

13	469	35	1620	57	884	79	725
14	1174	36	1601	58	2037	80	1963
15	693	37	298	59	1481	81	3979
16	1908	38	874	60	559	82	1090
17	135	39	618	61	490	83	245
18	277	40	2640	62	593	84	1194
19	596	41	5	63	1769	85	994
20	757	42	149	64	85	--	--
21	437	43	1034	65	2836	--	--
22	2230	44	2441	66	213	--	--

Table 6. 2: 4^{th} order and 5th order of Table 6.1

t 4	Values of t_4	t5	Values of t5
1	1576	**1**	2610
2	4149	**2**	4436
3	5827	**3**	8163
4	10071	**4**	11836
5	11836	**5**	15685
6	15280	**6**	17995
7	16860	**7**	22226
8	19572	**8**	28257
9	23827	**9**	32346
10	28257	**10**	39856
11	31886	**11**	46147
12	34467	**12**	53223
13	40751	**13**	58996
14	48262	**14**	67374
15	53223	**15**	80106
16	56160	**16**	91190
17	61565	**17**	98692
18	69815		
19	82822		
20	91190		
21	97698		

Table: 6.3: Parameter Estimates and their 4 and 5 order

Data Set	*Order*	$\hat{a}$	$\hat{b}$
Table 6. 2	**4**	2.14497	0.000064
	5	1.764372	0.000069

Table 6.4: Estimation of a,b and Specification of b0 and b1

Data Set	Order	Estimation of a	Estimation of b	b_0	b_1
SYS2 Data	4^{th} Order	2.14497	0.000064	0.000039	0.000089
	5^{th} Order	1.764372	0.000069	0.000044	0.000094

Table 6.5: SPRT Analysis for SYS2 Data Set with 4TH ORDER

Data Set	T	N(t)	R.H.S of equation (5.4) Acceptance region (≤)	R.H.S of Equation (5.5) Rejection Region(≥)	Decision
SYS2 Data Set	1576	**1**	-1.787145	3.720333	**REJECT**
	4149	**2**	-1.675536	4.290204	
	5827	**3**	-1.635921	4.654817	
	10071	**4**	-1.641618	5.574667	
	11836	**5**	-1.685636	5.964108	
	15280	**6**	-1.837911	6.750975	
	16860	**7**	-1.936470	7.128539	
	19572	**8**	-2.147392	7.808531	

Table 6.6: SPRT Analysis for SYS2 Data Set with 5TH ORDER

Data Set	T	N(t)	R.H.S of equation (5.4) Acceptance region (≤)	R.H.S of Equation (5.5) Rejection Region(≥)	Decision
SYS2 Data Set	2610	**1**	-1.962747	4.261248	**CONTINUE**
	4436	**2**	-1.941419	4.681990	
	8163	**3**	-1.995008	5.548850	
	11836	**4**	-2.167014	6.443052	
	15685	**5**	-2.469493	7.458772	
	17995	**6**	-2.711442	8.122341	
	22226	**7**	-3.277352	9.475104	
	28257	**8**	-4.385980	11.805201	
	32346	**9**	-5.370590	13.734575	
	39856	**10**	-7.786046	18.265995	
	46147	**11**	-10.578594	23.367578	
	53223	**12**	-14.841301	31.057102	
	58996	**13**	-19.476023	39.365311	
	67374	**14**	-28.722235	55.880861	
	80106	**15**	-51.318955	96.147834	
	91190	**16**	-84.494810	155.210457	
	98692	**17**	-118.137655	215.087869	

Conclusion

This paper shows the proposed model of detecting exponential reliable software using an SPRT with Ordered statistics. The SPRT illustrated on SYS2 data set with 4th and 5th ordered statistics, which indicate that the model is performing well in arriving at a decision. The detection results shows that a decision of rejection for 4th order data set SYS2 and continue for 5th order data set SYS2 at various time instant of the data. Therefore we can come to an early conclusion of predicting a reliable / unreliable of software.

Acknowledgment

We gratefully acknowledge the support of Department of Computer Science and Engineering, Acharya Nagarjuna University and St Mary Group of Institutions which providing necessary facilities to carry out the research work.

References

[1]Goel, A., "Software Reliability Models: Assumptions, Limitations, and Applicability," IEEE Trans. Software Eng.Vol. SE-11, No. 12 (April 1985), pp. 375-386.

[2]B. Srinivasa Rao, Dr. R.Satya Prasad, Dr. R.R.L.Kantham, “Monitoring Software Reliability using Statistical Process Control :An Ordered Statistics Approach”, (IJCA) Vol.1, No.7, October- 2011.

[3] John D.Musa; “Software Quality and Reliability Basics”; AT&T Bell Laboratories. CH 2468-7/87/0000/014,1987 IEEE.

[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] GOEL, A.L and OKUMOTO, K. (1979). "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.

[7] Balakrishnan. N., Clifford Cohen. A., "Order Statistics and Inference", Academic Press, INC. page 13. 1991.

[8]K.Ramchand H Rao, Dr. R.Satya Prasad, Dr. R.R.L.Kantham, "Assessing Software Reliability Using SPC: An Order Statistics Approach", (IJCSEA) Vol.1, No.4, August 2011.

[9]GOVINDARALUJU. (1981). "The sequential statistical analysis of hypothesis testing, point and interval estimation and decision theory". American series in mathematical and management sciences.

[10]GHOSH. (1970). "Sequential tests of statistical hypotheses. "Addison- Wesley.

[11]HOLM, S. (1985). "Sequential likelihood ratio tests. Sequential methods in statistics" Banach center publications. 16, 193-232.

[12] D. Haritha ,Dr. R. Satya Prasad. "Detecting Reliable Software Using SPRT: An Order Statistics Approach". IJCST Vol. 4, ISSue 2, AprIl - June 2013