CHAPTER 5

DETECTION OF RELIABLE PARETO SOFTWARE USING SPRT

Rapid growth of software usage enforces us to access the software reliability, a critical task in the development of a software system. Various models are adopted to assess the software reliability. It requires considerably less number of observations when compared with the other existing testing procedures. In Classical Hypothesis testing volumes of data is to be collected and then the conclusions are drawn, which may need more time. But, Sequential Analysis of Statistical science could be adopted in order to decide upon the reliability or unreliability of the developed software very quickly. The procedure adopted for this is, Sequential Probability Ratio Test (SPRT). It is designed for continuous monitoring. The likelihood based SPRT proposed by Wald is very general and it can be used for many different probability distributions. In the present chapter we propose the performance of SPRT on 11 different data sets using Pareto Type-IV and analyzed the results. The parameters are estimated using Maximum Likelihood Estimation method. The content of this chapter is published in the following journal and the details are furnished below.


5.1 Introduction

Sequential Probability Ratio Test (SPRT), which is usually applied in situations, requires a decision between two simple hypotheses or a single decision point. Wald’s (1947) SPRT procedure has been used to classify the software under test into one of two categories (e.g., reliable/unreliable, pass/fail, certified/noncertified) (Reckase, 1983). Wald's procedure is particularly relevant if the data is collected sequentially. Classical Hypothesis Testing is different from Sequential Analysis. In Classical Hypothesis testing, the number of cases tested or collected is fixed at the beginning of the experiment. In this method, the analysis is
made and conclusions are drawn after collecting the complete data. However, in
Sequential Analysis every case is analysed directly. The data collected up to that
moment is then compared with threshold values, incorporating the new information
taken from the freshly collected case. This approach makes one to draw conclusions
during the data collection, and ultimate conclusion can be reached at a much earlier
stage. Data collection can be terminated after few cases and decisions can be taken
quickly. This leads to saving in terms of cost and human life.

In the analysis of software failure data, either TBFs or failure count in a
given time interval is dealt with. 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 it is known that the probability equation of the stochastic
process representing the failure occurrences is given by a Homogeneous Poisson
Process with the expression

\[ P[N(t) = n] = \frac{e^{-\lambda t} (\lambda t)^n}{n!} \quad (5.1.1) \]

Stieber (1997) observes that, the application of SRGMs may be difficult and
reliability predictions can be misleading, if classical testing strategies are used.
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 chapter we consider the popular SRGM – a three parameter Pareto Type-IV
model and adopt the principle of Stieber(1997) in detecting unreliable software in
order to accept or reject the developed software. The theory proposed by
Stieber(1997) is presented in Section 5.2 for a ready reference. Extension of this
theory to the considered SRGM is presented in Section 5.3. Maximum Likelihood
parameter estimation method is presented in Section 5.4. Application of the decision
rule to detect unreliable software with reference to the SRGM-Pareto type IV is given in Section 5.5.

5.2 Wald’s Sequential Test for a Poisson Process

A. Wald, developed the SPRT at Columbia University in 1943. A big advantage of sequential tests is that they require fewer observations (time) on the average than fixed sample size tests. SPRT is widely used for statistical quality control in manufacturing processes. The SPRT for Homogeneous Poisson Processes is described below.

Let \( \{N(t), t \geq 0\} \) be a homogeneous Poisson process with rate ‘\( \lambda \)’. In this case, \( N(t) \) = number of failures up to time ‘\( t \)’ and ‘\( \lambda \)’ is the failure rate (failures per unit time). If the system is put on test (for example a software system, where testing is done according to a usage profile and faults are corrected) and that if we want to estimate its failure rate ‘\( \lambda \)’. We cannot expect to estimate ‘\( \lambda \)’ precisely. But we want to reject the system with a high probability if the data suggest that the failure rate is larger than \( \lambda_1 \) and accept it with a high probability, if it is smaller than \( \lambda_0 \). As always with statistical tests, there is some risk to get the wrong answers. So we have to specify two (small) numbers ‘\( \alpha \)’ and ‘\( \beta \)’, where ‘\( \alpha \)’ 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. ‘\( \beta \)’ 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. Wald’s classical SPRT is very sensitive to the choice of relative risk required in the specification of the alternative hypothesis. With the classical SPRT, tests are performed continuously at every time point \( t > 0 \) as additional data are collected. With specified choices of \( \lambda_0 \) and \( \lambda_1 \) such that \( 0 < \lambda_0 < \lambda_1 \), the probability of finding \( N(t) \) failures in the time span \((0, t)\) with \( \lambda_1, \lambda_0 \) as the failure rates are respectively given by

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

\[
P_0 = \frac{e^{-\lambda_0 t} \lambda_0^t N(t)}{N(t)!}
\]
The ratio $\frac{p_1}{p_0}$ at any time ‘t’ is considered as a measure of deciding the truth towards $\lambda_0$ or $\lambda_1$, 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 $\lambda_1$, in favour of $\lambda_0$ 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 to

$$\frac{p_1}{p_0} \geq A \quad (5.2.3)$$

$$\frac{p_1}{p_0} \leq B \quad (5.2.4)$$

$$B < \frac{p_1}{p_0} < A \quad (5.2.5)$$

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

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

Where ‘$\alpha$ ’ and ‘$\beta$ ’ 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

$$N_U(t) = at + b_2 \quad (5.2.6)$$
To accept the system to be reliable if $N(t)$ falls for the first time below the line

$$N_L(t) = at - b_1$$

(5.2.7)

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 equations (5.2.6) and (5.2.7) where

$$a = \frac{\lambda_1 - \lambda_0}{\log(\frac{\lambda_1}{\lambda_0})}$$

(5.2.8)

$$b_1 = \frac{\log(\frac{1-\alpha}{\beta})}{\log(\frac{\lambda_1}{\lambda_0})}$$

(5.2.9)

$$b_2 = \frac{\log(\frac{1-\beta}{\alpha})}{\log(\frac{\lambda_1}{\lambda_0})}$$

(5.2.10)

The parameters $\alpha, \beta, \lambda_0$ and $\lambda_1$ can be chosen in several ways. One way suggested by Stieber (1997) is

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

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

If $\lambda_0$ and $\lambda_1$ are chosen in this way, the slope of $N_Y(t)$ and $N_L(t)$ equals $\lambda$.

The other two ways of choosing $\lambda_0$ and $\lambda_1$ 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).
5.3 Sequential Test for Software Reliability Growth Models

In Section 5.2, 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

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

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

\[
m(t) = a \left[ 1 - \left( 1 + \left( \frac{t}{C} \right)^{-b} \right)^{-1} \right]
\]

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 \( \frac{P_1}{P_0} \leq B \)
\[ i.e., \quad \frac{e^{-m_1(t)}}{e^{-m_0(t)}} \left[ \frac{m_1(t)}{m_0(t)} \right]^{N(t)} \leq B \]

\[ i.e., \quad N(t) \leq \frac{\log(\frac{\beta}{1-\alpha}) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} \quad (5.3.1) \]

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

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

\[ i.e., \quad N(t) \geq \frac{\log(\frac{\beta}{1-\alpha}) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} \quad (5.3.2) \]

Continue the test procedure as long as

\[ \frac{\log(\frac{\beta}{1-\alpha}) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} < N(t) < \frac{\log(\frac{\beta}{1-\alpha}) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} \quad (5.3.3) \]

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:

\[ N(t) \leq \frac{\log(\frac{\beta}{1-\alpha}) + a \left[ 1 + \left( \frac{\ell}{c_0} \right)^{-b_0} - 1 + \left( \frac{\ell}{c_1} \right)^{-b_1} \right]}{\log a \left[ \left( \frac{1 + (\ell/c_0)^{-b_0}}{1 + (\ell/c_1)^{-b_1}} \right)^{b_1} \right]} \quad (5.3.4) \]
Rejection Region:

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

(5.3.5)

Continuation Region:

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

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

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

(5.3.6)

It may be noted that in the proposed model the decision rules are exclusively based on the strength of the sequential procedure \((\alpha, \beta)\) 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 (5.3.1), (5.3.2), (5.3.3) 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 5.4.
5.4 SPRT Analysis of Datasets

In this section, the developed SPRT methodology is shown for a software failure data which is of interval domain. We evaluate the decision rules based on the considered mean value function for Six different data sets borrowed from (Pham 2005), (Wood 1996) and five different data sets, borrowed from Pham (2006), Zhang et al. (2002), Ohba (1984a), Misra (1983) are evaluated. Based on the estimates of the parameter ‘b’ in each mean value function, we have chosen the specifications of $b_0 = b - \delta$, $b_1 = b + \delta$ and $c_0 = c - \delta$, $c_1 = c + \delta$ equidistant on either side of estimate of b obtained through a data set to apply SPRT such that $b_0 < b < b_1$ and $c_0 < c < c_1$.

**Assumptions:**

(i) $\delta = 0.8$ for calculation of $b_0$ and $b_1$.

(ii) $\delta = 0.1$ for calculation of $c_0$ and $c_1$ of phase 1 data, phase 2 data and Release 1 to Release 4 datasets.

(iii) $\delta = 2$ for calculation of $c_0$ and $c_1$ of Dataset #1a, Dataset #2a, Dataset #3a

(iv) $\delta = 4$ for calculation of $c_0$ and $c_1$ of Dataset #4a and Dataset #5a respectively.

The estimates are given in the following Table 5.4.1.
Table 5.4.1: Estimates of a, b, c & specifications of b0, b1,c0, c1

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Estimate of ‘a’</th>
<th>Estimate of ‘b’</th>
<th>b0</th>
<th>b1</th>
<th>Estimate of ‘c’</th>
<th>c0</th>
<th>c1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pham (2005)</td>
<td>36.772248</td>
<td>0.978993</td>
<td>0.178993</td>
<td>1.778993</td>
<td>9.541455</td>
<td>9.441455</td>
<td>9.641455</td>
</tr>
<tr>
<td>Phase 1 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pham (2005)</td>
<td>58.835596</td>
<td>0.978993</td>
<td>0.178993</td>
<td>1.778993</td>
<td>9.541455</td>
<td>9.441455</td>
<td>9.641455</td>
</tr>
<tr>
<td>Phase 2 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wood (1996)</td>
<td>123.844535</td>
<td>0.978352</td>
<td>0.178352</td>
<td>1.778352</td>
<td>9.144224</td>
<td>9.044224</td>
<td>9.244224</td>
</tr>
<tr>
<td>Release #1 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wood (1996)</td>
<td>158.153536</td>
<td>0.977674</td>
<td>0.177674</td>
<td>1.777674</td>
<td>8.74581</td>
<td>8.64581</td>
<td>8.84581</td>
</tr>
<tr>
<td>Release #2 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wood (1996)</td>
<td>83.720313</td>
<td>0.971698</td>
<td>0.171698</td>
<td>1.771698</td>
<td>5.978218</td>
<td>5.878218</td>
<td>6.078218</td>
</tr>
<tr>
<td>Release #3 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wood (1996)</td>
<td>60.600888</td>
<td>0.977674</td>
<td>0.177674</td>
<td>1.777674</td>
<td>8.74581</td>
<td>8.64581</td>
<td>8.84581</td>
</tr>
<tr>
<td>Release #4 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset #1a</td>
<td>66.190046</td>
<td>0.978993</td>
<td>0.178993</td>
<td>1.778993</td>
<td>9.541455</td>
<td>7.541455</td>
<td>11.541455</td>
</tr>
<tr>
<td>Dataset #2a</td>
<td>37.628137</td>
<td>0.973637</td>
<td>0.173637</td>
<td>1.773637</td>
<td>6.766191</td>
<td>4.766191</td>
<td>8.766191</td>
</tr>
<tr>
<td>Dataset #3a</td>
<td>59.481269</td>
<td>0.976195</td>
<td>0.176195</td>
<td>1.776195</td>
<td>7.953206</td>
<td>5.953206</td>
<td>9.953206</td>
</tr>
<tr>
<td>Dataset #4a</td>
<td>105.329506</td>
<td>0.985185</td>
<td>0.185185</td>
<td>1.785185</td>
<td>15.109772</td>
<td>11.109772</td>
<td>19.109772</td>
</tr>
<tr>
<td>Dataset #5a</td>
<td>200.558922</td>
<td>0.985185</td>
<td>0.185185</td>
<td>1.785185</td>
<td>15.109772</td>
<td>11.109772</td>
<td>19.109772</td>
</tr>
</tbody>
</table>

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 (5.3.4), (5.3.5) sequentially at each ‘t’ of the data set taking the strength \( (\alpha, \beta) \) as (0.05, 0.05).
### Table 5.4.2: SPRT Analysis for 11 Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$T$</th>
<th>$N(t)$</th>
<th>R.H.S. of Equation 5.3.4 Acceptance Region (⇐)</th>
<th>R.H.S. of Equation 5.3.5 Rejection Region (≥)</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pham (2005)</td>
<td>1</td>
<td>1</td>
<td>0.616561</td>
<td>2.181801</td>
<td>Accept</td>
</tr>
<tr>
<td>Phase 1 Data</td>
<td>2</td>
<td>1</td>
<td>1.610231</td>
<td>3.117998</td>
<td></td>
</tr>
<tr>
<td>Pham (2005)</td>
<td>1</td>
<td>3</td>
<td>1.294371</td>
<td>2.685788</td>
<td>Reject</td>
</tr>
<tr>
<td>Phase 2 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wood (1996)</td>
<td>1</td>
<td>16</td>
<td>3.097891</td>
<td>4.279688</td>
<td>Reject</td>
</tr>
<tr>
<td>Release #1 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wood (1996)</td>
<td>1</td>
<td>13</td>
<td>4.093627</td>
<td>5.218637</td>
<td>Reject</td>
</tr>
<tr>
<td>Release #2 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wood (1996)</td>
<td>1</td>
<td>6</td>
<td>3.132024</td>
<td>4.392929</td>
<td>Reject</td>
</tr>
<tr>
<td>Release #3 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wood (1996)</td>
<td>1</td>
<td>1</td>
<td>1.495721</td>
<td>2.873153</td>
<td>Accept</td>
</tr>
<tr>
<td>Release #4 Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset #1a</td>
<td>1</td>
<td>2</td>
<td>1.086669</td>
<td>2.450446</td>
<td>Reject</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>2.414109</td>
<td>3.742010</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>3.414343</td>
<td>4.710416</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>4.177357</td>
<td>5.444947</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>7</td>
<td>4.765048</td>
<td>6.006958</td>
<td></td>
</tr>
<tr>
<td>Dataset #2a</td>
<td>1</td>
<td>1</td>
<td>0.631835</td>
<td>2.187158</td>
<td>Accept</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1.609641</td>
<td>3.107547</td>
<td></td>
</tr>
<tr>
<td>Dataset #3a</td>
<td>1</td>
<td>3</td>
<td>1.123785</td>
<td>2.516498</td>
<td>Reject</td>
</tr>
<tr>
<td>Dataset #4a</td>
<td>1</td>
<td>6</td>
<td>0.962533</td>
<td>2.206964</td>
<td>Reject</td>
</tr>
<tr>
<td>Dataset #5a</td>
<td>1</td>
<td>9</td>
<td>2.108384</td>
<td>3.203745</td>
<td>Reject</td>
</tr>
</tbody>
</table>

From the Table (5.4.2) it is observed that a decision either to accept or reject the system is reached much in advance of the last time instant of the data.
5.5 Conclusion

The Table 5.4.2 of Interval domain data as exemplified for 11 Data Sets shows that Pareto Type IV model is performing well in arriving at a decision. Out of 11 Datasets the procedure applied on the model has given a decision of rejection for 8 datasets, acceptance for 3 datasets and continue for none at various time instant of the data. Phase1 is accepted at 2\textsuperscript{nd} instant of time, Release #4 is accepted at 1\textsuperscript{st} instance of time and Dataset #2a is accepted at 2\textsuperscript{nd} instance of time whereas remaining datasets are rejected at different instances of time. Therefore, by applying SPRT on data sets it can be concluded that we can come to an early conclusion of reliable or unreliable software.