CHAPTER 5
SOFTWARE RELIABILITY USING SPRT

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 / 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 5 data sets of Time domain data and analyzed the results. The parameters are estimated using Maximum Likelihood Estimation method.

5.1 INTRODUCTION

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

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 (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 popular
SRGM Weibull and adopt the principle of Stieber (1997) in detecting unreliable
software components in order to accept or reject the developed software. The
theory proposed by Stieber (1997) is presented in Section 2 for a ready reference.
Extension of this theory to the SRGM – Inflection S-shaped model is presented in
Section 3. Application of the decision rule to detect unreliable software with
respect to the proposed SRGM is given in Section 4. Analysis of the application
of the SPRT on 5 data sets and conclusions drawn are given in Section 5.

5.2 WALD'S SEQUENTIAL TEST FOR A POISSON PROCESS

The sequential probability ratio test (SPRT) 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 ‘\lambda’.
In our case, 
\[N(t) = \text{number of failures up to time 't'}\] and ‘\lambda’ 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 ‘\lambda’. We can not expect to estimate ‘\lambda’
precisely. But we want to reject the system with a high probability if our data suggest that the failure rate is larger than $\lambda_1$ and accept it with a high probability, if it’s 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.

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

$$Q_1 = \frac{e^{-\lambda_0 t} [\lambda_1 t]^{N(t)}}{N(t)!}$$  \hspace{1cm} (5.2.1)

$$Q_0 = \frac{e^{-\lambda_1 t} [\lambda_0 t]^{N(t)}}{N(t)!}$$  \hspace{1cm} (5.2.2)

The ratio $\frac{Q_1}{Q_0}$ 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 $t_1 < t_2 < t_3 < \ldots < t_K$ and the corresponding realizations $N(t_1), N(t_2), \ldots, N(t_K)$ of $N(t)$.

Simplification of $\frac{Q_1}{Q_0}$ gives

$$\frac{Q_1}{Q_0} = \exp(\lambda_0 - \lambda_1)t + \left(\frac{\lambda_1}{\lambda_0}\right)^{N(t)}$$

The decision rule of SPRT is to decide in favor of $\lambda_1$, in favor of $\lambda_0$ or to continue by observing the number of failures at a later time than ‘$t$’ according as $\frac{Q_1}{Q_0}$ 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 the test process with one more observation in failure data, according as

$$\frac{Q_1}{Q_0} \geq A$$  \hspace{1cm} (5.2.3)
\[ \frac{Q_1}{Q_0} \leq B \]  
\[ B < \frac{Q}{Q_0} < A \]  

(5.2.4)  
(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

\[ N_U(t) = a t + b_2 \]  

(5.2.6)

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

\[ N_L(t) = a t - b_1 \]  

(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

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

(5.2.8)

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

(5.2.9)

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

(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

\[ \lambda_0 = \frac{\lambda_1 \log(q)}{q - 1}, \quad \lambda_1 = q \frac{\lambda_1 \log(q)}{q - 1} \quad \text{where} \quad q = \frac{\lambda_1}{\lambda_0} \]

If \( \lambda_0 \) and \( \lambda_1 \) are chosen in this way, the slope of \( N_U(t) \) and \( N_L(t) \) equals \( \lambda \). The other two ways of choosing \( \lambda_0 \) and \( \lambda_1 \) are from past projects and from part of the data to compare the reliability of different functional areas.
5.3 SEQUENTIAL TEST FOR SRGM

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

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

Depending on the forms of \( m(t) \) we get various Poisson processes called NHPP.

For the Inflection S-shaped model the mean value function is given as

\[
m(t) = \frac{a}{1 + ce^{-bt}} (1 - e^{-bt}) \quad \text{where} \quad a > 0, b > 0
\]

We may write

\[
Q_1 = \frac{e^{-m_1(t)} [m_1(t)]^{N(t)}}{N(t)!
}
\]

\[
Q_0 = \frac{e^{-m_0(t)} [m_0(t)]^{N(t)}}{N(t)!}
\]

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.

Let \( P_0, P_1 \) be values of the NHPP at two specifications of \( b \) say \( b_0, b_1 \) where \( (b_0 < b_1) \) 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

\[
\frac{Q_1}{Q_0} \leq B
\]

i.e.,

\[
\frac{e^{-m_1(t)} [m_1(t)]^{N(t)}}{e^{-m_0(t)} [m_0(t)]^{N(t)}} \leq B
\]

\[
\log \left( \frac{\beta}{1 - \alpha} \right) + m_1(t) - m_0(t) \leq B
\]

i.e., \( N(t) \leq \frac{\log \left( \frac{\beta}{1 - \alpha} \right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} \) (5.3.1)
Decide the system to be unreliable and reject if

\[
\frac{Q_1}{Q_0} \geq A
\]

i.e., \(N(t) \geq \frac{\log \left( \frac{1 - \beta}{\alpha} \right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)}\) \hspace{1cm} (5.3.2)

Continue the test procedure as long as

\[
\frac{\log \left( \frac{\beta}{1 - \alpha} \right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)} \leq N(t) < \frac{\log \left( \frac{1 - \beta}{\alpha} \right) + m_1(t) - m_0(t)}{\log m_1(t) - \log m_0(t)}\]

Substituting the appropriate expressions of the respective mean value function – \(m(t)\) of Inflection S-shaped model we get the respective decision rules and are given in followings lines

Acceptance region:

\[
N(t) \leq \frac{\log \left( \frac{\beta}{1 - \alpha} \right) + a (1 + c) \left( e^{-b_1 t} - e^{-b_2 t} \right)}{\log \left( \frac{1 - e^{-b_1 t}}{1 + e^{-b_1 t}} \right) \left( \frac{1 + e^{-b_2 t}}{1 + e^{-b_2 t}} \right) - \log \left( \frac{1 - e^{-b_2 t}}{1 + e^{-b_2 t}} \right) \left( \frac{1 + e^{-b_1 t}}{1 + e^{-b_1 t}} \right)}
\]

(5.3.4)

Rejection region:

\[
N(t) \geq \frac{\log \left( \frac{1 - \beta}{\alpha} \right) + a (1 + c) \left( e^{-b_1 t} - e^{-b_2 t} \right)}{\log \left( \frac{1 - e^{-b_1 t}}{1 + e^{-b_1 t}} \right) \left( \frac{1 + e^{-b_2 t}}{1 + e^{-b_2 t}} \right) - \log \left( \frac{1 - e^{-b_2 t}}{1 + e^{-b_2 t}} \right) \left( \frac{1 + e^{-b_1 t}}{1 + e^{-b_1 t}} \right)}
\]

(5.3.5)

Continuation region:

\[
\frac{\log \left( \frac{\beta}{1 - \alpha} \right) + a (1 + c) \left( e^{-b_1 t} - e^{-b_2 t} \right)}{\log \left( \frac{1 - e^{-b_1 t}}{1 + e^{-b_1 t}} \right) \left( \frac{1 + e^{-b_2 t}}{1 + e^{-b_2 t}} \right) - \log \left( \frac{1 - e^{-b_2 t}}{1 + e^{-b_2 t}} \right) \left( \frac{1 + e^{-b_1 t}}{1 + e^{-b_1 t}} \right)} < N(t) < \frac{\log \left( \frac{1 - \beta}{\alpha} \right) + a (1 + c) \left( e^{-b_1 t} - e^{-b_2 t} \right)}{\log \left( \frac{1 - e^{-b_1 t}}{1 + e^{-b_1 t}} \right) \left( \frac{1 + e^{-b_2 t}}{1 + e^{-b_2 t}} \right) - \log \left( \frac{1 - e^{-b_2 t}}{1 + e^{-b_2 t}} \right) \left( \frac{1 + e^{-b_1 t}}{1 + e^{-b_1 t}} \right)}
\]

(5.3.6)

It may be noted that in the above 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 4.

5.4 SPRT ANALYSIS OF LIVE DATA SETS

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 Five different data sets of the above form,
borrowed from Pham (2006) and Lyu(1996). Based on the estimates of the
parameter ‘b’ in each mean value function, we have chosen the specifications of
\(b_0 = b - \delta, \ b_1 = b + \delta\) equidistant on either side of estimate of b obtained through a
Data Set to apply SPRT such that \(b_0 < b < b_1\). Assuming the value of \(\delta = 0.0025\),
the choices are given in the following table.

Table 5.4.1: Estimates of \(a, b\) & Specifications of \(b_0, b_1\) for Time domain

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Estimate of ‘a’</th>
<th>Estimate of ‘b’</th>
<th>(b_0)</th>
<th>(b_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>33.239615</td>
<td>0.003212</td>
<td>0.000712</td>
<td>0.005712</td>
</tr>
<tr>
<td>DS2</td>
<td>33.498685</td>
<td>0.006139</td>
<td>0.003639</td>
<td>0.008639</td>
</tr>
<tr>
<td>DS3</td>
<td>23.298053</td>
<td>0.003595</td>
<td>0.001095</td>
<td>0.006095</td>
</tr>
<tr>
<td>DS4</td>
<td>25.301345</td>
<td>0.003057</td>
<td>0.000557</td>
<td>0.005557</td>
</tr>
<tr>
<td>DS5</td>
<td>31.404848</td>
<td>0.020966</td>
<td>0.018466</td>
<td>0.023466</td>
</tr>
</tbody>
</table>

Using the selected \(b_0, b_1\) and subsequently the \(m_0(t), m_1(t)\) for the model, we
calculated the decision rules given by Equations 5.3.4 and 5.3.5, sequentially at
each ‘t’ of the data sets taking the strength (\(\alpha, \beta\)) as \((0.05, 0.2)\). These are
presented for the model in Table 5.4.2. The following consolidated table reveals
the iterations required to come to a decision about the software of each Data Set.
Table 5.4.2: SPRT analysis for 5 data sets of Time domain data

<table>
<thead>
<tr>
<th>Data Set</th>
<th>T</th>
<th>N(t)</th>
<th>Acceptance region ($\leq$)</th>
<th>Rejection Region ($\geq$)</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>30.02</td>
<td>1</td>
<td>1.454440</td>
<td>3.420660</td>
<td>Acceptance</td>
</tr>
<tr>
<td>DS2</td>
<td>9</td>
<td>1</td>
<td>-0.065094</td>
<td>4.611109</td>
<td>Acceptance</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>2</td>
<td>1.896209</td>
<td>6.755759</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>3</td>
<td>3.569635</td>
<td>8.603840</td>
<td></td>
</tr>
<tr>
<td>DS3</td>
<td>10</td>
<td>1</td>
<td>-0.203087</td>
<td>2.123273</td>
<td>Rejection</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>2</td>
<td>0.334327</td>
<td>2.695469</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>3</td>
<td>1.077341</td>
<td>3.489403</td>
<td></td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>4</td>
<td>1.676664</td>
<td>4.132460</td>
<td></td>
</tr>
<tr>
<td></td>
<td>58</td>
<td>5</td>
<td>2.452731</td>
<td>4.969152</td>
<td></td>
</tr>
<tr>
<td>DS4</td>
<td>5.5</td>
<td>1</td>
<td>-0.333444</td>
<td>1.386230</td>
<td>Rejection</td>
</tr>
<tr>
<td></td>
<td>7.33</td>
<td>2</td>
<td>-0.241196</td>
<td>1.482474</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>1</td>
<td>-5.621252</td>
<td>10.865743</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.7</td>
<td>2</td>
<td>-4.969968</td>
<td>11.745192</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.5</td>
<td>3</td>
<td>-3.519429</td>
<td>13.742045</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.2</td>
<td>4</td>
<td>-2.209376</td>
<td>15.598067</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>5</td>
<td>-0.939089</td>
<td>17.455348</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>6</td>
<td>0.326249</td>
<td>19.374240</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14.8</td>
<td>7</td>
<td>1.039668</td>
<td>20.492543</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15.7</td>
<td>8</td>
<td>1.383852</td>
<td>21.042869</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17.1</td>
<td>9</td>
<td>1.902975</td>
<td>21.887672</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20.6</td>
<td>10</td>
<td>3.116462</td>
<td>23.942933</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>11</td>
<td>4.184340</td>
<td>25.868015</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25.2</td>
<td>12</td>
<td>4.536099</td>
<td>26.532026</td>
<td></td>
</tr>
</tbody>
</table>
From the above table, a decision of either to accept, reject the system or continue is reached much in advance of the last time instant of the data.

5.5 CONCLUSION.

The above consolidated table shows that Inflection S-shaped model as exemplified for 5 Data Sets indicate that the model is performing well in arriving at a decision. The model has given a decision of acceptance for 2 Data Sets i.e DS1 & DS2, a decision of rejection for 2 Data Sets i.e DS3 & DS4 and Continue for 1 Data set i.e DS5. Therefore, we may conclude that, applying SPRT on data sets we can come to an early conclusion of reliability / unreliability of software.