INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

Detection of Reliability using Sprt: S-Shaped Models

S.Chitti Babulu[1], Dr. R. Satya Prasad[2], Dr. K.Raja Sekhara Rao[3]

[1]Research Scholar, Department of CSE, JNTUK, Kakinada, India.

[2]Professor, Dept. CSE, ANU, Guntur, India.

[3]Professor, Dept. CSE, KLEF, Guntur, India.

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150100092

Received: 28 January 2026; Accepted: 02 February 2026; Published: 16 February 2026

ABSTRACT

Traditional hypothesis testing may cause delay to take important decisions, as it depends on collecting a lot of evidence before conclusions are drawn. An alternative method for assessing software reliability is the sequential analysis, the Sequential Probability Ratio Test (SPRT). SPRT provides the mechanism of continuous monitoring which made it possible to attribute reliable or unreliable software rapidly. A framework of SPRT is proposed by Wald for different probability distributions. This paper proposed to use SPRT on ungrouped software failure data of six datasets collected from literature categorized as ungrouped with two popular S-shaped models. Real Valued Genetic Algorithm is proposed for parameter estimation to assess the performance evaluation.

Keywords*:* SRGMs, S-shaped models, SPRT, RVGA.

INTRODUCTION

An alternative to classical hypothesis testing is the SPRT proposed by Wald (1947) that evaluates data sequentially. comparing accumulated evidence with predefined thresholds allows continuous assessment and early termination of testing. This is very effective for decisions which involves binary classification such as reliability or unreliability. Time Between Failures (TBF) and Failure counts (FC) are the failure patterns commonly found in software reliability analysis. A process is modelled as a Homogeneous Poisson Process when failures occur with a constant rate and follows Poisson distribution which describes the occurrence of failures. Assuming a constant failure rate λ following a Poisson distribution the stochastic behaviour of software is given as follows which enables to gain insight.

$$Q[O(x) = j] = \frac{e^{-\lambda x}(\lambda x)^x}{j!} \qquad (1.1)$$

The limitations of traditional testing methods were highlighted by Stieber (1997) for Software Reliability Growth Models (SRGMs) for reliability predictions which leads to misleading decisions. He proposed the use of SPRT for failure data analysis. The present work applies two popular S-shaped models to assess reliability of software following the principle proposed by Stieber. Application of this principle for software acceptance or rejection is explained in section 2. It is extended to the considered NHPP-based SRGMs in section 3. Section 4 presents the decision rule being applied for identifying unreliable software. Section 5 describes the proposed Real-Valued Genetic Algorithm for parameter estimation, with estimated model parameters given in Section 6. Section 7 evaluates the models on the datasets and discusses their effectiveness in detecting unreliable software.

Wald's Sequential Test

The SPRT, developed by A. Wald, originated from wartime research and was initially classified due to its military applications. Its key advantage over fixed-sample tests is the reduced average number of observations required, leading to efficient decision-making. Consequently, SPRT is widely used in statistical quality control

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

and manufacturing as shown in homogeneous Poisson processes, it offers an effective framework for sequential hypothesis testing with optimized resource use.

Let {O(x), x ≥0} be a HPP with rate ‘λ’. In our case, O(x) = count of failures up to ‘x’ time and ‘λ’ is the rate of failure. Suppose a system is placed on test and aimed to estimate its ‘λ’. it cannot expect to exactly estimate. The system is rejected with a likelihood high if data under consideration recommend that the rate of failure is greater than λ_1. if it is smaller than λ_0 accept it with probability high. As statistical tests, have some risk getting the answers misleading. Two small numbers ‘u’ and ‘v’ are specified representing the probability of falsely refusing, if $\lambda \le \lambda_0$, considered as "producer’s" risk. ‘v’ is the probability of falsely accepting, if $\lambda \ge \lambda_1$, considered as the “consumer’s” risk.

The relative risk choice essential in the description of the alternative hypothesis will influence highly the Wald’s SPRT. Continuously at each point of time x > 0 classical SPRT tests are performed, as data is collected additionally. With choices of λ_0 and λ_1 such that $0 < \lambda_0 < \lambda_1$, the probability of finding O(x) failures in the time span (0, x) with λ_1, λ_0 as failure rates is given by (Murali mohan et al., 2013)

$$Q_1 = \frac{e^{-\lambda_1 x}(\lambda_1 x)^{o(X)}}{o(X)!} \qquad (2.1)$$

$$Q_0 = \frac{e^{-\lambda_0 x}(\lambda_0 x)^{o(X)}}{o(X)!} \qquad (2.2)$$

At any time ’x’, $\frac{Q_1}{Q_0}$ decides the truth towards λ_0 or λ_1, given a sequence of time, say $x_1 < x_2 < x_3 < \ldots < x_k$ and the corresponding realizations $O(x_1), O(x_2), \ldots, O(x_k)$ of $O(x)$. $\frac{Q_1}{Q_0}$ simplification gives

$$\frac{Q_1}{Q_0} = \exp(\lambda_0 - \lambda_1)\, x + (\frac{\lambda_1}{\lambda_0})$$

The rule of decision is to decide in favor of λ_0 , in favor of λ_1, or to continue by observing the number of failures later than 'x'. As $\frac{Q_1}{Q_0}$ is greater than or equal to a constant say R, less than or equal to a constant say S or in between the constants R and S. The product under consideration is decided as continue, reliable, unreliable of the test process with another observation in data.

$$\frac{Q_1}{Q_0} \ge R \qquad \frac{Q_1}{Q_0} \le S \qquad S < \frac{Q_1}{Q_0} < R$$

The values of R and S are approximately taken as

$$R \cong \frac{1-v}{u}, \qquad S \cong \frac{v}{1-u}$$

Where ‘u’ and ‘v’ are the probabilities of risk. A test that minimizes both the (u) and (v) errors as much as possible is considered as good. The procedure in general is to fix the v error and then choose a critical region to maximize or minimize the error. A easier version of the above decision is to reject, as unreliable if above the line, $O(x)$ falls for the first.

$$O_u(x) = y.x + z_2 \qquad (2.6)$$

if $O(x)$ falls for the first below the line, accept as reliable

$$O_L(x) = y.x - z_1 \qquad (2.7)$$

To proceed the test with another on $(x, O(x))$ as the random graph of $[(x, O(x))]$ is between the two linear

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

boundaries given by equations (2.6) and (2.7) (Satya prasad et al., 2013).

$$y = \frac{\lambda_1 - \lambda_0}{log\left(\frac{\lambda_1}{\lambda_0}\right)} \quad z_1 = \frac{log\left(\frac{1-u}{v}\right)}{log\left(\frac{\lambda_1}{\lambda_0}\right)} \quad z_2 = \frac{log\left(\frac{1-v}{u}\right)}{log\left(\frac{\lambda_1}{\lambda_0}\right)}$$

The parameters chosen in multiple ways i.e u, v, λ_0 and λ_1. Stieber suggested as,

$$\lambda_0 = \frac{\lambda log(d)}{d-1} \quad \lambda_1 = d\frac{\lambda log(d)}{d-1} \quad \text{where } d = \frac{\lambda_1}{\lambda_0}$$

the slope of $O_u(x)$ and $O_L(x)$ = λ, If λ_0 and λ_1 are above.

NHPP based SRGMs

SRGMs grounded in Non-Homogenous Poisson Processes (NHPP) have demonstrated considerable efficacy in real-world software reliability engineering, as evidenced by Haidry et al. (2023). The main problem with NHPP models being used in reliability lies in a suitable Mean Value Function (MVF). As all the models are based on assumptions, various NHPP SRG models are constructed, and their parameter estimation is a huge task. Various algorithms were proposed in the literature to facilitate model parameters estimation. Huang et al. (2022) proposed debugging models with imperfection acknowledging the possibility that fault removal may not be immediate and may introduce new faults.

In SRGMs the cumulative count of software failures by time ‘x’ is given by $\{O(x), x > 0\}$. The MVF representing the cumulative count of expected software failures detected up to a given time is represented by $m(x)$. Predicting and assessing software reliability during testing phases is a critical concept in SRGMs. The failure intensity $\lambda(x)$ is proportional to residual fault content. These include models like:

Delayed S-shaped

The Delayed model, based on NHPP theory, describes software error detection as an S-shaped growth curve, with slow initial detection, rapid discovery in the middle phase, and a gradual slowdown as remaining errors become harder to find. The MVF is given as $m(x) = y(1-(1+zx)e^{-zx})$ Were, $y > 0, z > 0$. ‘z’ is the error detection rate per error in state of steady. ‘y’ is the count of initial faults (Pełka, 2012).

Inflection S-shaped

Ohba’s inflection model (1984) characterizes an accelerating fault detection rate during testing, controlled by an inflection parameter that reflects the proportion of detectable faults while limiting unrealistic exponential growth. Its mean value function is $m(x) = y\left(\frac{1-e^{-zx}}{1+ce^{-zx}}\right)$,where ‘x’ is time, $m(x)$ is the expected cumulative failures by time ‘x’, ‘y’ is the total expected failures, ‘z’ is the detection rate of failure, and ‘c’ is the factor of inflection indicating the transition from rapid to slower fault detection.

Sequential Test for Software Reliability Growth Models

In Section 2, the expected value of $O(x) = \lambda x$, which represents the average count of failures in time x, is referred to as the MVF of the Poisson process. On the other hand, if we consider a Poisson process with a funnction $m(x)$as its MVF, the probability equation of such a process is given by:

$$Q[O(x) = j] = \frac{[m(x)]^j}{j!}e^{-m(x)}, j = 0,1,2,\ldots$$

various Non-homogeneous Poisson processes (NHPP) are obtained based on the form of $m(x)$. For a delayed

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

S-shaped model it is given as (Yamada *et al.*, 1984).

$$Q_1 = \frac{e^{-m_1(x)}[m_1(x)]^{O(x)}}{O(x)!}, \qquad Q_0 = \frac{e^{-m_0(x)}[m_0(x)]^{O(x)}}{O(x)!}$$

Where, $m_1(x)$, $m_0(x)$ are values of the MVF at specified sets of its parameters representing reliable and unreliable software respectively. Let Q_0, Q_1 be values of the NHPP at two specifications of 'z' say $z_{0,}$ z_1, where $(z_{0,} < z_1)$. It can be shown that for the model $\mathrm{m}(x)$ at z_1 is greater than that at z_0 . Symbolically $m_0(x) < m_1(x)$. Then the SPRT procedure is as follows:

if $\frac{Q_1}{Q_0} \leq S$, the system is considered as reliable and is accepted.

i.e., $$O(x) \leq \frac{\log\left(\frac{v}{1-u}\right)+m_1(x)-m_0(x)}{log m_1(x)-log m_0(x)} \tag{4.1}$$

if $\frac{Q_1}{Q_0} \geq R$, the system is considered as unreliable and is rejected.

i.e., $$O(x) \geq \frac{\log\left(\frac{1-v}{u}\right)+m_1(x)-m_0(x)}{log m_1(x)-log m_0(x)} \tag{4.2}$$

Continue the test as long as

$$O(x) < \frac{\log\left(\frac{v}{1-u}\right)+m_1(x)-m_0(x)}{log m_1(x)-log m_0(x)} < O(x) < \frac{\log\left(\frac{1-v}{u}\right)+m_1(x)-m_0(x)}{log m_1(x)-log m_0(x)} \tag{4.3}$$

By substituting the respective MVF $m(x)$for the delayed S-shaped model, the corresponding conclusion rules are obtained, as presented in the following lines.

Acceptance region:

$$O(x) \leq \frac{\log\left(\frac{v}{1-u}\right)+a\left[(1+z_0x)e^{-z_0x}-\left((1+z_1x)e^{-z_1x}\right)\right]}{log\left[\frac{1-(1+z_1x)e^{-z_1x}}{1-(1+z_0x)e^{-z_0x}}\right]} \tag{4.4}$$

Rejection region:

$$O(x) \geq \frac{\log\left(\frac{1-v}{u}\right)+a\left[(1+z_0x)e^{-z_0x}-\left((1+z_1x)e^{-z_1x}\right)\right]}{log\left[\frac{1-(1+z_1x)e^{-z_1x}}{1-(1+z_0x)e^{-z_0x}}\right]} \tag{4.5}$$

Continuation region:

$$\frac{\log\left(\frac{1-v}{u}\right)+a\left[(1+z_0x)e^{-z_0x}-\left((1+z_1x)e^{-z_1x}\right)\right]}{log\left[\frac{1-(1+z_1x)e^{-z_1x}}{1-(1+z_0x)e^{-z_0x}}\right]} < O(x) < \frac{\log\left(\frac{v}{1-u}\right)+a\left[(1+z_0x)e^{-z_0x}-\left((1+z_1x)e^{-z_1x}\right)\right]}{log\left[\frac{1-(1+z_1x)e^{-z_1x}}{1-(1+z_0x)e^{-z_0x}}\right]} \tag{4.6}$$

Similar rules are derived for Inflection S-shaped model. The judgement rules are determined solely by the strength of the sequential procedure $(u, v)$and the corresponding MVFs $m_0(x)$and $m_1(x)$. Stiber described that the judgement rules become conclusion lines when MVF is linear in 'x' passing through origin, that is, $m(x) = \lambda x$. In that sense equations (4.1), (4.2), (4.3) can be regarded as generalizations to the Stieber decision procedure. The applications of these for software failure data are obtained with analysis. Let

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

- H_0: The S-shaped model adequately fits the observed failure data
- H_1: The S-shaped model fails to adequately fit the observed failure data.

For each dataset, the cumulative inter-failure time is plotted against:

- Acceptance boundary
- Rejection boundary

Decision Rules:If the cumulative curve crosses the accept boundary, H_0 is accepted. If it crosses the reject boundary, H_0 is rejected. If it remains between both boundaries, the test is inconclusive and sampling continues.

METHODOLOGY

Data Collection

Software failure data analysis is a critical process in understanding and improving software reliability. This analysis is essential for understanding the behaviour of software systems, enabling targeted fault detection and correction strategies. For the improvement of quality, the analysis of reliability growth and robust data sets provide valuable insights for effective use of strategies.

Parameter Estimation:

SRGMs rely on techniques of parameter estimation like Maximum Likelihood Estimation, Least Square Estimation to predict software reliability. The methods proposed by Koren and krishna (2021) and Latha Shanmugam et al. (2013) necessitate the collection of failure data to estimate the unknown parameters. The potential constraints related to unimodality, continuity and the existence of derivatives within complex likelihood functions is highlighted by Costa et al. (2010). LSE proves more suitable for small to medium sample sizes as suggested by Mahmood et al. (2022). Genetic Algorithm (GA) based parameter estimation for Hyper-Geometric Distribution is proposed by Minohara and Tohma (1995). Zhen's (2020) application of Particle Swarm Optimization (PSO) for parameter estimation have demonstrated limitations.

Genetic Algorithm (GA):

Jiang (2006) and McMinn (2011) proposed parameter estimation can be effectively conceptualized as a search within the solution space. The incorporation of GA into software reliability modelling is a potentially advantageous approach. Lopes et al. (2024) demonstrated the application of a refined RVGA coupled with empirical data through the S-shaped models elucidates both the parameter estimation process and the resultant assessments. This integrated methodology offers a robust framework for improving software reliability predictions.

Real Valued Genetic Algorithm

For a large set of diverse problems, the GAs flexibility offers a versatile approach to optimization. Their strength towards quality solutions is ease of implementation, effectiveness and rapid convergence in large search spaces. A significant adaptation of the traditional GA for continuous variables is RVGA tailored for optimization problems. RVGAs uses real numbers to avoid the encoding and decoding in handling continuous values with binary strings.

Real Valued Genetic Algorithm Parameter Estimation

A RVGA is used to estimate the SRG models parameters. The efficacy of this approach is demonstrated through Table 6.1 presenting the parameters estimated for six data sets available from the literature. This research contributes software reliability prediction through the application of RVGA and the comparative analysis of the

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

delayed and inflection models under consideration.

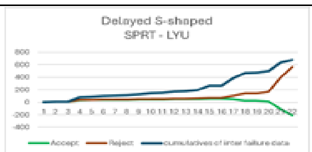
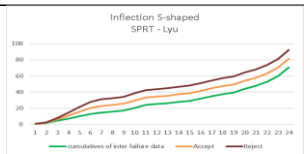
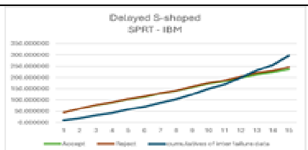
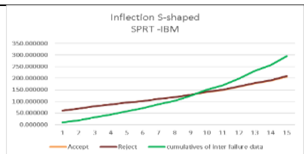
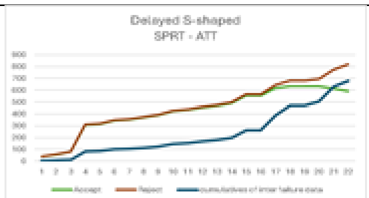
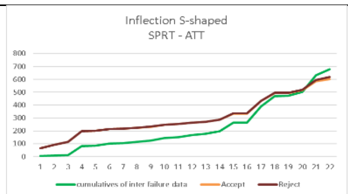
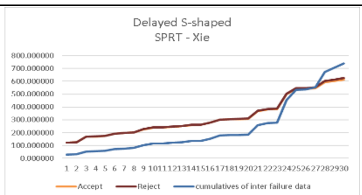
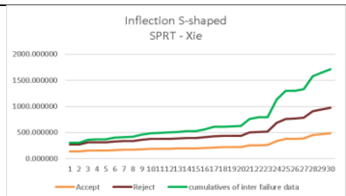
Table 6.1: Estimated Parameters

Data Set	Delayed		Inflection		
	Y	**Z**	**y**	**Z**	**c**
DS1 (LYU)	69.994993	0.110755	70.000000	0.180372	15.962549
DS2 (IBM)	295.999928	0.187389	296.000000	0.357895	9.544106
DS3 (AT&T)	680.00000	0.109179	679.999127	0.276183	15.826329
DS4 (Xie)	737.891028	0.063255	729.866440	0.134056	25.262465
DS5 (NTDS)	249.999901	0.097633	250.000000	0.159340	18.053581
DS6 (SONATA)	1831.982958	0.112970	1831.928582	0.176066	16.183552

SPRT-Based Analysis

In this section, we evaluate the conclusion rules using the selected MVF across six different data sets obtained from Pham (2006), Xie et al. (2002), and SONATA (Ashoka, 2010). Based on the estimated value of the parameter z from each MVF, the specifications $z_0 = z - \theta$and $z_1 = z + \theta$, were chosen and positioned equidistantly on either side of the estimated z obtained from the dataset. These choices ensure that $z_0 < z < z_1$and enable the application of the SPRT. The decision rules are computed using Equations 4.4 and 4.5 assuming a value of $\theta$for each model using the corresponding mean value functions $m_0(x)$and $m_1(x)$. These rules are evaluated sequentially at each x in the datasets, with the strength parameters (w, v)set to $(0.05, 0.2)$. The resulting decision rules for the models are presented graphically in Table 7.1.

Table 7.1: SPRT analysis for 6 sets

Data Set	Delayed S-shaped	Inflection S-shaped
LYU		
IBM		
AT&T		
Xie		

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

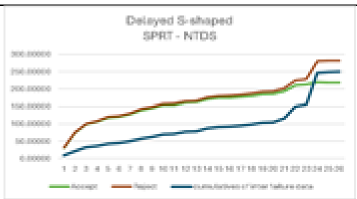
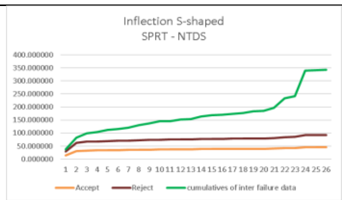
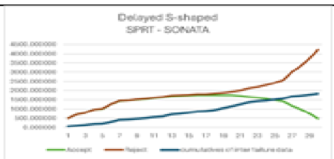
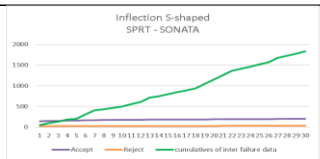
NTDS		
SONATA		

Fig: Graphical representation of SPRT.

The results presented in the above table clearly indicate that the decision regarding system acceptance or rejection is made significantly earlier than the final observation time. To statistically validate the applicability of this model, the SPRT is employed. This section presents a comprehensive SPRT-based evaluation of the Delayed and Inflection S-shaped SRGM using six real-world software failure datasets: LYU, IBM, ATT, NTDS, SONATA, and Xie.

7.2 Comparative Summary of SPRT Results

Dataset	**Delayed**		**Inflection**	
	SPRT Decision	**Suitability**	**SPRT Decision**	**Suitability**
LYU	Inconclusive / Weak	Poor	Inconclusive	Weak
IBM	Accept H_0	Good	Accept H_0	Good
ATT	Accept H_0	Excellent	Accept H_0	Excellent
NRDS	Accept H_0 (Borderline)	Reasonable	Reject H_0	Not Suitable
SONATA	Reject H_0	Not Suitable	Reject H_0	Not Suitable
Xie	Reject H_0	Not Suitable	Reject H_0	Not Suitable

CONCLUSION

SPRT-based results show that the Delayed S-shaped SRGM performs well for IBM and ATT datasets, which exhibit delayed learning with mild acceleration, but fails to capture late-stage failure surges in SONATA and Xie, while only partially fitting the inconsistent LYU data. The Inflection S-shaped model also fits IBM and ATT well, showing significant inflection behavior and early SPRT acceptance. However, it is rejected for Xie, SONATA, and NTDS due to excessive acceleration or erratic failures, and yields inconclusive results for LYU owing to the absence of a clear inflection point.

REFERENCES

1. Ashoka. M., (2010). “Sonata software limited Data Set”, Bangalore.
2. Costa. E. O, Pozo. A, and Vergilio. S. R, "A Genetic Programming Approach for Software Reliability Modeling," IEEE Transactions on Reliability, vol. 59, no. 1, pp. 222–230, 2010.
3. Haidry, A.I, Jameel. T, Naveed. A, Riaz. R and Razzaq. L, "Software Reliability Analysis Using Prediction Models," 2023 20th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Bhurban, Murree, Pakistan, 2023, pp. 315-320.
4. Huang. Y.-S., Chiu. K.-C, and Chen. W.-M, “A software reliability growth model for imperfect debugging,” Journal of Systems and Software, vol. 188, p. 111267, Jun. 2022.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue I, January 2026

5. Jiang. H, "Can the Genetic Algorithm Be a Good Tool for Software Engineering Searching Problems?" 30th Annual International Computer Software and Applications Conference (COMPSAC'06), Chicaco, IL, USA, 2006, pp. 362-366, doi: 10.1109/COMPSAC.2006.123.
6. Koren. I, and Krishna. C. M, "Software Fault Tolerance," Fault-Tolerant Systems, pp. 161–202, 2021, doi: https://doi.org/10.1016/b978-0-12-818105-8.00015-2.
7. Latha Shanmugam, Lilly Florence, T. Srilatha. "Evaluation of Software Reliability Models.", International Conference on Current Trends in Advanced Computing ICCTAC 2013. ICCTAC, 1 (June 2013), 54-57.
8. Lopes. T, Tomazella. V. L. D, Leão. J, Ramos. P. L, and Louzada. F, "Statistical Inference for Generalized Power-Law Process in repairable systems," Journal of Computational and Applied Mathematics, vol. 445, p. 115799, Aug. 2024.
9. Mahmood. A, Hameed. K, Zameer. A, Abdullah. A, and Sajid. S, "Review of Software Reliability through Prediction Models," 2022 International Conference on Recent Advances in Electrical Engineering & Computer Sciences, Islamabad, Pakistan, pp. 1-5, 2022,
10. McMinn. P, "Search-Based Software Testing: Past, Present and Future," 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops, Berlin, Germany, 2011, pp. 153-163, doi: 10.1109/ICSTW.2011.100.
11. Minohara, S., & Tohma, Y. (1995). Parameter estimation of discrete probability distributions using genetic algorithms. IEEE International Conference on Systems, Man and Cybernetics, Proceedings, pp. 1815–1820.
12. Murali Mohan, S. Satya Prasad, R. and Krishna Mohan, G. “Exponential Software reliability using SPRT: MLE”, IOSR Journal of Computer Engineering (IOSR-JCE). e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 13, Issue 2 (Jul. - Aug. 2013), PP 36-41.
13. Pełka R. Software Reliability Growth Models. Computer Science and Mathematical Modelling. (2012);0(10/2012):19-29.
14. Pham. H., (2006). “System software reliability”, Springer.
15. Satya Prasad, R. Murali Mohan, S. and Krishna Mohan, G. “A Two Step Approach for parameter estimation of software reliability”, Elixir *International Journal: Computer Science and Engineering*. (60) july, 2013, [16341-16344]. ISSN: 2229–712X.
16. Stieber, H.A. (1997). “Statistical Quality Control: How To Detect Unreliable Software Components”, Proceedings the 8th International Symposium on Software Reliability Engineering, 8-12.
17. Wald. A., 1947. “Sequential Analysis”, John Wiley and Son, Inc, New York.
18. Xie, M., Goh. T.N., Ranjan.P., “Some effective control chart procedures for reliability monitoring” - Reliability engineering and System Safety 77, 143 -150¸ 2002.
19. Yamada, S., Ohba, M. and Osaki, S., (1984). "S-shaped software reliability growth models and their applications," IEEE Transactions on Reliability, Vol. R-33.
20. Zhen. L., Liu. Y., Dongsheng. W. and Wei. Z, "Parameter Estimation of Software Reliability Model and Prediction Based on Hybrid Wolf Pack Algorithm and Particle Swarm Optimization," in IEEE Access, vol. 8, pp. 29354-29369, 2020.