OPEN ACCESS

International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

An Empirical Analysis of Software Reliability using Reliability Analysis Algorithm

Gayathry G, Thirumalaiselvi R

Abstract*: Reliability is the fundamental aspect of a software system that cannot be ignored and hard to measure. Two major elements namely hardware and software need to be measured to evaluate the reliability of software system. The work that is already existing give focus on measuring the reliability of software alone. With little consideration in measuring the reliability of hardware. The present work focuses on computation of hardware reliability and software reliability together. The aim is to propose an algorithm to develop a model for estimation of reliability. The algorithm is named as reliability analysis algorithm. Using comparison criteria developed model is compared with other two traditional models. The result of this study shows that the developed model can be used to measure and predict the reliability with high degree of accuracy.

Keywords: Mean Value Function, Failure Intensity, Parameter Estimation, Model Criteria, Software Reliability Growth Models.

I. INTRODUCTION

Over the past decade, the deployment of software products has grown more dramatically. It is an integral part of informational society. On concerning the development of software, the basic concept of software reliability and its measurement is attracting and receiving a lot of attention of researchers. The software reliability methodology that are already available tries to provide quantitative measures to predict or estimate the reliability. Most of the industries say for example telecommunication and banking perform their daily routines and activities using computers. This dependent quality on computer increases the failure rate. To address this issue, reliability become a major concern in our modern world. The failure due to the software products can affect the outcome. Not only the developers, the users also need a reliable software product. There are various methodologies followed to develop software. The factors that need to be evaluated are software failure time and the method for testing coverage and when to stop the testing process and to release the software. The reliability analysis can be done at various stages of development life cycle. The analysis performed during the process of software development, as an attempt to evaluate whether it satisfies the specified requirements. To detect and remove latent faults testing process need to be carried out during the final stage of software development process.

Manuscript published on 30 September 2019
* Correspondence Author

Gayathry G*, Computer Science, Bharathiar University, Coimbatore, India.

Thirumalaiselvi R, Computer Science, Govt. Arts College (Men), Nandanam, Chennai, India.

II. RELATED WORK

Kiranjit Kaur et al [6] discussed about software metrics and how it improves the reliability and quality of the software product. Shelbi et al [4] estimate the reliability of computational systems based on newly developed model. Harminder et al [1] identifies that Maximum Likelihood Estimates gives the better results as compared to Least Square Estimates for Goodness of Fit and predictive validity of Models. Maximum Likelihood Estimator is better for prediction of Reliability Growth Models. Anni Princy et al, [7] discussed about introducing Gompertz testing effort in reliability models. Analysis is based on two dimensions such as testing time and testing coverage. Mohammed Ibraigheeth et al, [8] reviewed reliability prediction methods that estimate failures in software. It also discusses the strength and limitation of prediction methods.

III. PROPOSED ALGORITHM

The newly proposed algorithm provides guideline for the users who wants to evaluate the software reliability.

Reliability Analysis Algorithm

Input: Past-Failure Data
Output: Expression for mean value, failure rate and reliability
Define a time interval t
Estimate a value
Estimate b value
For each time interval t
Begin
Calculate mean value for sw failure
$m(t) = a(1-e^{-bt})$
Calculate mean value for hw failure
$m(t)= e^{-\lambda hw\ (t)}$
Obtain the expression of Mean Value
$m(t)=e- ^{-\lambda hw\ (t)} + a(1-e^{-bt})$
Calculate failure rate for sw as $\lambda\ (t) = abe^{-bt}$
Calculate failure rate for hw as $\lambda\ (t) =\lambda_{hw}\ (t)$
Obtain the expression of Failure Rate
$\lambda\ (t) = \lambda_{hw}\ (t) + abe^{-bt}$
End
Obtain the reliability expression for software as
$R(x|t) = e^{-(\lambda hw * x)} + e^{-a\ (}e^{-bt} - e^{-b(t+x))}$

IV. COMPARISON CRITERIA

The commonly used metrics for model comparisons are discussed in table I.

Table I: Criteria and its Description [3]

Criteria	Description	Expression
Bias	Sum of the difference between the estimated curve, and the actual data.	$\sum_{i=1}^{k} \frac{(m_i - m(t_i))^2}{n}$
Mean Square Error	Deviation between the predicted values with the actual observations	$\sum_{i=1}^{k} \frac{(m_i - m(t_i))^2}{k-p}$
Mean Absolute Error	Absolute value is used to measure the deviation	$\sum_{i=1}^{k} \frac{\lvert(m_i - m(t_i))\rvert}{k-p}$
Mean Error of Prediction	Sums the absolute value of the deviation between the actual data and the estimated curve	$\sum_{i=1}^{k} \frac{\lvert(m_i - m(t_i))^2\rvert}{k-p+1}$
Accuracy of Estimation	Measures the difference between the estimated numbers of all errors with the actual number of all detected errors	$\left\lvert \frac{M_a - a}{M_a} \right\rvert$
Noise	Measure the error	$\sum_{i=1}^{k} \left\lvert \frac{(\lambda(t_i) - \lambda(t_{i-1}))}{\lambda(t_{i-1})} \right\rvert$
Predictive-Ratio Risk	Measures the distance of model estimates from the actual data against the model estimate	$\sum_{i=1}^{k} \frac{(m(t_i) - m_i)}{m(t_i)}$
Variance	Standard deviation of the prediction bias	$\sqrt{\frac{\sum_{i=1}^{n} (m_i - m(t_i) - Bias)^2}{k-1}}$
Root Mean Square Prediction Error	It is a measure of the closeness with which the model predicts the observation	$\sqrt{Variance^2 + Bias^2}$
R Square	Measure how successful the fit is in explaining the variation of the data	$\frac{\sum_{i=1}^{k} (m_i - m(t_i))^2}{\sum_{i=1}^{n} (m_i - \sum_{j=1}^{k} \frac{m_j}{n})^2}$
Sum of Squared Errors	Measure the errors	$\sum_{i=1}^{k} (m_i - m(t_i))^2$
Theil Statistic	It is the average deviation percentage over all periods with regard to the actual values	$\sqrt{\frac{\sum_{i=1}^{n} (m_i - m(t_i))^2}{\sum_{i=1}^{n} m_i^2}}$

V. DATA COLLECTION

In order to measure the reliability, hardware failure data considered by Gayathry et al [5] is used in this paper. From the dataset, the value of hardware failure rate calculated as $e^{-0.225753*t}$. For software failures failure data considered by Mohd Anjum et al [3] is used. These datasets are used to evaluate and compare the model. Failure dataset is represented in Table II.

Table II: Software Failure Dataset [3]

Week	Cumulative Faults	Week	Cumulative Faults
1	15	11	149
2	35	12	157
3	60	13	173
4	74	14	179
5	94	15	182
6	102	16	184
7	114	17	185
8	134	18	187
9	139	19	191
10	148	20	192

The proposed model is compared with Musa-Okumoto and Gompertz model. The table III represent the estimated values of parameters for the newly proposed model and other two traditional model.

Table III: Maximum Likelihood Estimation of Parameter

Parameter/ Model	Musa Okumoto	Gompertz	Proposed Model
a	113	191.78	215.76
b	0.230	0.242	0.108
c	-	-0.059	-

VI. ANALYSIS AND EXPERIMENTAL RESULTS

The comparison values are represented in table IV.

Table IV: Comparative results of different SRGM

Criteria/Model	Musa Okumoto	Gompertz	Proposed Model
AE	0.0014	0.0003	5.92E-05
Noise	0.0300	0.6664	0.1517
Rsq	0.9697	0.9725	0.9925
TS	0.2706	0.2346	0.0042
PRR	2.31E-06	1.59E-06	6.48E-07
RMSPE	0.3980	0.1579	0.0031
MAE	0.2552	0.1837	0

Retrieval Number C5712098319/2019©BEIESP
DOI: 10.35940/ijrte.C5712.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

When the Rsq value is very close to one, the model fits well with data. The value of Rsq for the proposed model is 0.9925 which is closer to 1. So, the proposed model best fit to the data. The TS value is 0.0042 which is closer to 0, it shows high prediction accuracy of the model. The graphical representation of MAE for the proposed model, Musa Okumoto, Gompertz are shown in the below Fig.1.

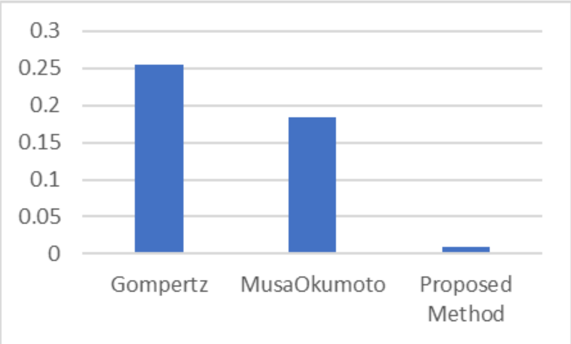

Fig. 1. Comparison of MAE

From the above table and graphs, MAE value is low when compared to traditional models Musa Okumoto and Gompertz model. Hence, it has been revealed that proposed model is effective and gives a highly predictive skill for the given software failure data.

VII. CONCLUSION

This study has proposed a reliability analysis algorithm. The proposed model is validated on failure data sets and analyses are done using comparison metrics. Also, it has been proved empirically that our proposed model is a valid indicator for reliability, which is considered as important software quality attributes. Analysis done through the newly developed algorithm has shown major improvement in estimation of reliability. From the table IV it is clear that the proposed algorithm creates a new path for effective reliability measurement.

REFERENCES

1. Harminder Pal Singh Dhami, Vaibhav Bansal.," Analysis of Software Reliability Growth Models for Quantitative Evaluation of Software Reliability and Goodness of Fitness Metrics", International Journal of Science and Research (2017) ISSN (Online): 2319-7064.
2. Hoang Pham., "A generalized fault-detection software reliability model subject to random operating environments", Vietnam J Computer Sci (2016) 3:145–150 DOI 10.1007/s40595-016-0065-1.
3. Mohd. Anjum, Md. Asraful Haque, Nesar Ahmad "Analysis and Ranking of Software Reliability Models Based on Weighted Criteria Value ", I.J. Information Technology and Computer Science, 2013, 02, 1-14.
4. Shelbi Joseph, Akhil P V, Seetha Parameswaran, "Reliability Estimation of Open Source Software based Computational Systems ", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 2, February 2017.
5. ThirumalaiSelvi R, Gayathry G, "A Model for Reliability Estimation Using Inter Failures Time Data", International Journal of Computer Sciences and Engineering, Vol.6, Issue 8, Aug.2018.
6. Kiranjit Kaur and Sami Anand," Review on Software and Hardware Reliability and Metrics ", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 5, May 2013.
7. Anni Princy B, Sridhar S,"Measuring Software Reliability and Release Time Using SRGM Tool ", International Journal Of Scientific Research And Education ,Volume 2, Issue 5, Pages 785-796 May 2014 ISSN (e): 2321-7545
8. Mohammad Ibraigheeth , Syed Abdullah Fadzli,"Software Reliability Prediction In Various Software Development Stages ",Journal of Theoretical and Applied Information Technology, Vol.96. No 7, Apr.2018. ISSN: 1992-8645.

AUTHORS PROFILE

Ms.G.Gayathry is working as Assistant professor in the Department of Computer Science, Mar Gregorios Arts and Science College, Chennai. She is pursuing her Ph.D in the area of software engineering from Bharathiar University, India. She has 14 years of teaching experience

Dr. (Mrs.) R.Thirumalaiselvi is currently the Research Supervisor and Assistant Professor in the Department of Computer Science, Govt. Arts College (Men) (Autonomous), Chennai. She has over 20+ years of experience in various arts and science colleges and as a research supervisor. She is guiding many PhD students registered under various universities.