A Methodology for SOFTWARE RELIABILITY BASED ON STATISTICAL MODELING

[1]Avinash seekoli, [2]Dr.Y. Srinivas, [3]Dr.P. AnnanNaidu

Ph.D scholar, professor, Assistant professor

avinash.seekoli@gmail.com, syarrama@gitam.edu, annannaidu@cutm.ac.in

Abstract--Reliability is one of the quantifiable quality features of the software. Software reliability growth models (SRGMs) are used to assess the reliability achieve at different test times based on statistical learning models. Conventional time based SRGMS may not be accurate enough in all situations and such models cannot identify errors in small and medium sized applications. Numerous traditional reliability measures are used to test software errors during application development and testing. In the software testing and maintenance phase, however, new errors are taken into account in real time in order to determine the reliability estimate. In this article, we suggest using the Weibull model as a computational approach to solving the problem of software reliability modeling. In the anticipated model, a new distribution model is projected to develop the reliability estimation method. We compute the model developed and balance its presentation through additional popular software reliability increase models commencing the literature. Our test consequences demonstrate that the planned Model is greater to S-shaped Yamada, comprehensive Poisson, NHPP.

Keywords: Software reliability, Software Reliability Growth Models, Testing, Statistical dependencies, Accuracy.

1. Introduction

Software-based systems play a predominant role in this modern world. In the future, the dependency on software systems will increase exponentially. In such a situation, these software systems should work properly. The expectation of a user community of any kind of software system is only one that is nothing more than software that is of very good quality. The meaning of quality in this context is: "The software should do what it should do, and it should not do what it should not do." According to this quality meaning, both Dos and Don should be concentrated.

To better understand software excellence, the various factors that authority software superiority should be examined. The factors affecting software quality are usability, performance, maintainability, reliability, etc. In order to distribute the software with elevated excellence, we need to improve the level of quality factors. In this view, software quality can be improved by increasing the dependability of the software. The present occupation examines the reliability of software under various software quality factors. The reliability of the software can survive uttered as the chance that it will effort successfully in a certain background used for a assured age of point.

Early Software Reliability Models (SRGM) represents the association between failure time and cumulative number of faults detect so far. “Many of these SRGMs have been proposed as parametric [1–14] and non-parametric [15–18] models in order to estimate future failure time of occurrence and assess the growth of software reliability during the test phase. Traditional SRGMs are on the basis of the principle that the mean value depends on the model follows either exponential growth [1, 3], or S-shaped growth [2, 11] or both [4–8]" [23].

The main purpose of SRGM is to build software quality at a low cost in a sensible period through beloved dependability. In process of reliability is single of the agent capacities. . The reliability of the software is essentially definite as the possibility of a predictable over quantify operation time interval. "It would be helpful to know the probability of failure free operation of a software system for a specified period of time" [10].

Several software models are used to improve the software measure of reliability and it can be classified into two levels namely prediction and estimation models. "These two techniques are based on the observation and compilation of failure data and the evaluation with statistical hypothesis. At first, using the prediction model, software reliability can be predicted initially in development step and improvements can be initiated to advance reliability" [10]. The other is to approximate the factor standards based on calculated or experimental data containing chance information.

Software dependability models are mainly of two kinds: one is with density defect [8] and second one is with increasing software reliability [9]." First kind of model see models so as to make an effort to predict software reliability based on design parameters and use implementation property such as nested loops, number of rows, input / output Estimate the number of software errors present"[23]."The second types, software reliability growth models, are applied on models that try to predict the reliability of software based on test data"[23]. "These models seek to show the relationship between fault detection data and well-known mathematical function such as Exponential and logarithmic functions" [23]. The suitability of these models depends on the level of correlation between the tests

In general," these models symbolize an increasing quantitative move toward to the extent of the software request" [10]. "Mainly the software reliability focused on results like faults and failures ignore the software development process. However, the complexity is reduced and an appropriate plan is made and new methods are introduced which are applied in some classes. So, we have to choose the correct model that suits our particular case" [10]. "In addition, the modeling results cannot be believed and applied blindly" [10]. in the present article, we suggest a dependability model base on software prediction.

this article is well thought-out like this: next section-II deals through nhpp, in section iii the data sets considered, section iv deals with at least square estimate, in section v, the software reliability model based on Weibull distribution is highlighted. conclusion of article in section vi.

2. Non homogeneous Poisson process (NHPP)

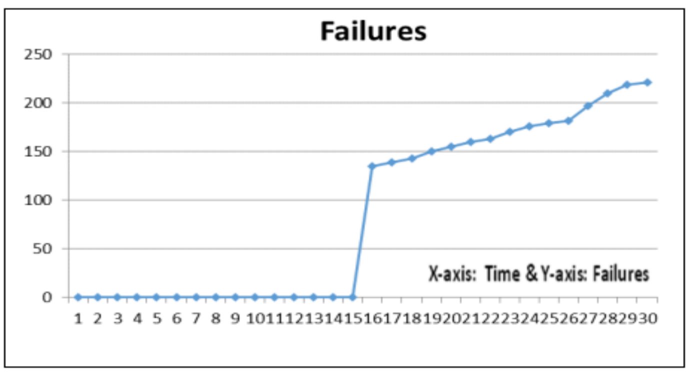
a non-homogeneous stochastic Poisson procedure is a progression using the limitation λ (t) it specifies several stochastic developments, which included forecasting forecasts for a sure value. the present process is definitely an extension of this model here λ (t) can be a random process. the non-homogeneous poission process model consists of determining an average significance task to indicate until a certain time the projected digit of failures practiced.

3. Data set

In this case study, we organize the model on KC2 dataset from NASA and the failures are projected accurately. The failure data so obtained is tested for correctness using Specificity, Sensitivity and F-measure. The original dataset of KC2 and its taxonomy process by Gaussian Mixture Model are existing below

Test Time (Weeks)	Estimated Failures
1	75
2	81
3	86
4	90
5	93
6	96
7	98
8	99
9	100
10	100
11	100
12	115
13	120
14	123
15	130
16	135
17	139
18	143
19	150
20	155
21	160
22	163
23	170
24	176
25	179
26	182
27	197
28	210
29	219
30	221

Table 1: Data Set



Graph-1: Failures in the given dataset

4. Least square estimation method

least squares parameter estimation involves discovering unknown variables during a certain solution. the function of knowledge is vital. it's a quite common approach the best method is minimizing the sum of squares of residuals, it is being the difference between an observed value and model provided the fitted value. the problem of least squares is occurred in regression analysis. During testing the statistical inference to failure data is obtained. the formula which is specified through least square evaluation is

y=a+bx where y is dependent variable,x is experimental variable,

Table 2. Least square estimation

Dataset #	a	B
1	31.524466	0.029012
2	23.655141	0.026521

5. Software reliability model based on Weibull distribution and Experimentation

In this method we have developed a new statistical method based on weilbul distribution and the formula for the estimate the PDF is presented below.

$$\boldsymbol{f(x)} = \sum_{x=0}^{1} \frac{\mathbf{1}}{\mathbf{2}\pi} \boldsymbol{e}^{-\left(\frac{x-a}{b}\right)(\mu-\sigma)} \qquad (\mathbf{1})$$

The values of a,b are estimated using the least square method and the values of μ and σ are estimated from the dataset.

5.1 Experimentation

Two data sets are considered for the experimentation. Each of the error values are given to the model to estimate the probability density functions.

Determination of fitness rate. to calculate fitness, calculate A regularized RMSE (Root Mean Square Error) is selected. it is also identified as standard error and standard regression error. A Root mean square error measurement better fit means nearer to zero. To calculate standardized RMSE see the below equation

$$\text{NRMSE} = \frac{1}{n}\sqrt{\left[\frac{1}{n}\sum_{i}^{n}(xi - yi)^2\right]} \qquad (2)$$

where n: is the no of data points.
x_i: ith point of observed data i.e orginal data
y_i: PDF

Table 3. Estimation Failures

Test Time (Weeks)	Estimated Failures	PDF Values
1	75	0.02186
2	81	0.02124
3	86	0.03264
4	90	0.03787
5	93	0.04882
6	96	0.05267
7	98	0.05989
8	99	0.06176
9	100	0.06878
10	100	0.06776
11	100	0.07251
12	115	0.07682
13	120	0.07987
14	123	0.07876
15	130	0.08962
16	**135**	**0.008182**
17	139	0.04156
18	143	0.05687
19	150	0.09247
20	155	0.09848
21	160	0.098415
22	163	0.09967
23	170	0.010892
24	176	0.010898
25	179	0.035678
26	182	0.05464
27	197	0.07856
28	210	0.06872
29	219	0.07876
30	221	0.09652

The methodology is tested against efficiency based on MSE and metrics such as sensivity and specificity. “We calculate and balance the goodness of fit (GoF) act of the proposed model using NRMSE” [22]. to calculate the square of the distinction among the real and expected values NRMSE is used. The smaller NRMS Eindicate the smallest adjustment error and better performance.

Table 4. NRMSE for test data

Root Mean Square Error Normalized Model	MSE	
	RMSE	Normalized RMSE
Yamada	22.7066	0.5677
Poisson 23.1365	23.1365	0.5784
NHPP	23.7988	0.5950
Raleigh Distribution	24.2316	0.5897
Proposed method	16.342	0.4123

Sensitivity, Specificity, and F-measure are used to test the efficiency of the model. And the values obtained are tabulated below in Table

Table 5. Sensitivity, specificity and F-measure

True Postive	False positive	Specificity	Sensitivity	F-measure
0.96	0.21	0.93.7	0.91	0.93
0.80	0.25	0.89	0.87	0.79

6. CONCLUSION

Software reliability is an important sophistication to quantifies the operational profile of computer systems. We proposed a software reliability growth model based on Weibull distribution. to estimating and monitoring software reliability the model is mainly used to calculate quality of software. Based on interval domain data developed the maximum likelihood estimates the parameters to find the equations. we conclude that our approach to estimation and the control chart are giving a positive recommendation based on mse and metrics based on sensitivity and specificity. this is an efficient method for validation and is convenient for practitioners in software reliability. in the present work the methodology is better than the procedure adopts by xie et al [2002]. Therefore, for an early detection of software failures we may conclude that the present model is the most suitable choice

References

[1] A. L. Goel and K. Okumoto, “Time-dependent error-detectionrate model for software reliability and other performance measures,” IEEE Transactions on Reliability, vol. 28, no. 3, pp.206–211

[2] S. Yamada and S. Osaki, “Software Reliability Growth Modeling:Models and Applications,” IEEE Transactions on Software Engineering, vol. SE-11, no. 12, pp. 1431–1437, 1985.

[3] P. K. Kapur and R. B. Garg, “A software reliability growthmodel for an error removal phenomenon,” Software EngineeringJournal, vol. 7, no. 4, pp. 291–294, 1992

[4] A. L. Goel, “Software reliability models: assumptions, limitations,and applicability,” IEEE Transactions on Software Engineering, vol. 11, no. 12, pp. 1411–1423, 1985.

[5] Y.Srinivas,Y.Vamsidhar, “software reliability growth model based on pareto type iii distribution,” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 6, pp 2694-2698, 2013

[6] R. Subburaj, G. Gopal, and P. K. Kapur, “A software reliability growth model for Vital Quality Metrics,” South African Journal of Industrial Engineering, vol. 18, no. 2, pp. 93–108, 2007.

[7] S. Ramasamy and G. Govindasamy, “A software reliabilitygrowth model addressing learning,” Journal of Applied Statistics,vol. 35, no. 9-10, pp. 1151–1168, 2008.

[8] R. Subburaj, G. Gopal, and P. K. Kapur, “A software reliabilitygrowth model for estimating debugging and the learningindices,” International Jounal of Performability Engineering, vol.8, no. 5, pp. 539–549, 2012.

[9] S. Ramasamy and A. M. J. Muthu Kumaran, “Dynamically weighted combination of fault - Based Software ReliabilityGrowth Models,” Indian Journal of Science and Technology, vol.9, no. 22, Article ID 93967, 2016.

[10] Y. Srinivas, Y.Vamsidhar,“Software Reliability Enhancement By Using Effective NHPP Models”, International Journal of Computer Science And Technology, vol 4(4), pp 312-314, 2013.

[11] S. Yamada, J. Hishitani, and S. Osaki, “Software-reliabilitygrowth with a Weibull test-effort: a model and application,”IEEE Transactions on Reliability, vol. 42, no. 1, pp. 100–105, 1993.

[12] C.-Y. Huang, S.-Y. Kuo, and M. R. Lyu, “An assessment oftesting-effort dependent software reliability growth models,”IEEE Transactions on Reliability, vol. 56, no. 2, pp. 198–211, 2007.

[13] P. K. Kapur, D. N. Goswami, A. Bardhan, andO. Singh, “Flexiblesoftware reliability growth model with testing effort dependentlearning process,” Applied Mathematical Modelling, vol. 32, no.7, pp. 1298–1307, 2008.

[14] J. Xu and S. Yao, “Software reliability growth model withpartial differential equation for various debugging processes,”Mathematical Problems in Engineering, Article ID2476584,Art.ID 2476584, 13 pages, 2016.

[15] P. Roy, G. S. Mahapatra, P. Rani, S. K. Pandey, and K. N.Dey, “Robust feed forward and recurrent neural network baseddynamic weighted combination models for software reliabilityprediction,” Applied Soft Computing, vol. 22, pp. 629–637, 2014.

[16] Y.-S. Su and C.-Y. Huang, “Neural-network-based approachesfor software reliability estimation using dynamic weightedcombinationalmodels,” Journal of Systems and Software, vol. 80,no. 4, pp. 606–615, 2007.

[17] I. Lakshmanan and S. Ramasamy, “An Artificial Neural-Network Approach to Software Reliability Growth Modeling,”in Proceedings of the 3rd International Conference on RecentTrends in Computing, ICRTC 2015, pp. 695–702, March 2015.

[18] S. Ramasamy and I. Lakshmanan, “Application of artificialneural network for software reliability growth modeling withtesting effort,” Indian Journal of Science and Technology, vol. 9,no. 29, Article ID 90093, 2016.

[19] M. Xie and M. Zhao, “On some reliability growth models withsimple graphical interpretations,” Microelectronics Reliability,vol. 33, no. 2, pp. 149–167, 1993.

[20] M. Xie, “Software Reliability Modeling, World Scientific,” Singapore Asia, 1991.

[21] A.Wood, “Predicting software reliability,” Computer, vol. 29,no.11, pp. 69–77, 1996.

[22] R. Subburaj, “Software Reliability Engineering, McGraw-Hill Professional,”New Delhi, India, 2015.

[23] Zainab ALRahamneh, Mohammad Reyalat, Alaa F. Sheta, Sulieman Bani-Ahmad, Saleh Al- Oqeili. "A New Software Reliability Growth Model: Genetic-Programming-Based Approach" , Journal of Software Engineering and Applications, 2011

[24] S. Yamada,“Software Reliability Modeling: Fundamentals andApplications,” Springer-Verlag, Tokyo, Japan, 2014.