

TESTING EFFORT BASED SOFTWARE RELIABILITY ASSESSMENT INCORPORATING FRF AND CHANGE POINT

Rajat ARORA

*Research Scholar, Department of Operational Research, University of Delhi.
arorarajat87@yahoo.com*

Anu G. AGGARWAL

*Professor, Department of Operational Research, University of Delhi.
anuagg17@gmail.com*

Received: March 2019 / Accepted: August 2019

Abstract: In today's World, to meet the demand of high quality and reliable software systems, it is imperative to perform comprehensive testing and debugging of the software code. The fault detection and removal rate may change over time. The time point after which the rates are changed is termed as the change point. Practically, the failure count may not coincide with the total fault count removed from the system. Their ratio is measured by Fault Reduction Factor (FRF). Here, we propose a Weibull testing effort dependent Software Reliability Growth Model with logistic FRF and change point for assessing the failure phenomenon of a software system. The models have been validated on two real software fault datasets. The parameters are estimated using Least squares and various criteria are employed to check the goodness of fit. The comparison is also facilitated with the existing models in literature to demonstrate that proposed model has better performance.

Keywords: SRGM, NHPP, FRF, Testing Effort, Change Point, Weibull Testing Effort Function.

MSC: 90B05, 90B06, 90C11.

1. INTRODUCTION

With rapid advances in computer technology, the software development has undergone a revolutionary change. Nowadays in mechanized environment, we are

increasingly becoming software dependent. The systems using software are used in diverse fields including medicine, education, data-processing, military, forecasting, real-time control systems and the list is endless. Hence, a high quality software is in great demand. The major challenge faced is the inescapability of defects in the software.

Reliability is one of the significant physiognomy of software excellence as it quantifies the fault content in the software during its development. SRGM are used to quantify the software reliability. These models clearly show relationship between the cumulative fault count at a time moment and the testing time. Various NHPP models in literature evaluate the failure phenomenon of software systems. Initially, researchers and developers did not consider the conditions affecting the software in testing and operational phases. Hence, these models were not that accurate in analyzing the failure process. With the increase in demand for reliable software in all fields, the developers and researchers have moved their focus towards the factors affecting the whole software development process.

Various factors affecting the process are: diverse testing environment and changing strategies, testing team's skill, time difference between detection of fault and its removal, addition of new faults, fault dependency, relationship between fault and failure, introduction of new technologies etc. Along with these, the new testing techniques and test cases significantly affect the fault detection rate (FDR) of the system and thus, it may not remain constant throughout the testing process. In previous studies researchers have assumed that every fault has equal probability of getting detected (Huang [10]; Huang and Lin [6]) but in reality the process of detecting faults is affected by the testing efficiency, the available resources, fault density etc. Therefore, FDR keeps on changing corresponding to these conditions. The time moment at which the FDR curve changes its direction is referred to as Change point. Lin & Huang introduced this concept with testing effort (Huang and Lin [16]). The occurrence of Change-point is a result of variation in testing environment.

In recent studies, researchers have incorporated these factors into SRGMs to predict and assess more accurately. Among them FRF is one of the significant factors first introduced by Musa (1987). As per Musa work FRF is defined as the percentage of net number of faults removed to the number of failures experienced (Musa et al. [19]). He took constant FRF. Practically, FRF is affected by various factors viz, resource allocation, fault dependency, environment, time lag in removing faults, learning process, error generation etc. Pachauri defined FRF as S-shaped function (Dhar et al. [21]). In this study, we consider logistic-type FRF.

Another key factor is Testing effort, expressed as the number of test cases, time spent on testing, testers employed etc. Allocating the resources properly is very critical for developing reliable software by minimizing the potential faults in the system. Therefore a developer needs to control the consumption of resources. This is highly correlated to the reliability and should be studied as time dependent factor (Huang and Kyo [7]). They are defined in the form of functions known as Testing Effort Functions (TEFs). Kapur, Goswami, & Gupta, (2004) Yamada, Hishitani, & Osaki, (1993) and Musa, (1987) discussed TEF in their models

(Goswami et al. [12]; Musa et al. [19]; Hishitani et al. [24]).

To address the issue of developing a more accurate SRGM with better goodness of fit we propose a new model incorporating TEF, Change point and FRF. In this paper, we model the fault detection/removal process corresponding to Testing effort function modeled by Weibull-curve. It incorporates Logistic FRF and change-point in perfect debugging environment. The results of the proposed model are validated on two real software failure datasets. Rest of the paper is organized as follows the next section develops the literature work corresponding to the factors incorporated in the model. Section three discusses the proposed model followed by its validation on fault datasets, and comparison with the models existing in literature is given in section four. At the end, the paper is concluded and future scope is presented.

2. LITERATURE REVIEW

Here, we will discuss the past research work carried out to assess the reliability growth of software systems.

2.1. Testing Effort

In literature, consumption of testing resources has been modelled using different functions as listed in Table 1. The notations used are defined as follows:

N : Total amount of testing effort available

b, δ : Shape parameters

m : Scale parameter

A : Constant

ρ : Rate of testing-effort consumption

2.2. Change Point

Considering change points helps to improve the accuracy of reliability models in managing software failure process. Zou, (2003) used change point concept for analysis software reliability when inter failure times are modelled by Weibull distribution (Zou [26]). Zhao & Wang, (2007) discussed the change point problems in modelling software and hardware failure phenomenon (Wang and Zhao [25]). X. Li, Xie, & Ng, (2010) performed sensitivity analysis on release time of a software system considering growth model with TEF and change points (Li et al. [15]). Inoue & Yamada, (2011) compared the failure phenomenon of software system before and after change in testing environment (Inoue and Yamada [11]).

Expressions	Testing-Effort Function(W)	Reference
$N(1 - e^{-bt})$	Exponential	Yamada et al. (1993)[24], Kapur, Gupta, Shatnawi and Yadavalli(2006)[13]
$N(1 - e^{(-\frac{b}{2})t^2})$	Rayleigh	Yamada et al.(1993)[24] Kapur, Gupta, Shatnawi and Yadavalli(2006)[13]
$N(1 - e^{-bt^m})$	Weibull	Yamada et al. (1993)[24] Kapur, Gupta, Shatnawi and Yadavalli(2006) [13]
$\frac{N}{1+Ae^{-\alpha t}}$	Logistic	Huang, Kuo, & Chen(1997)[8](Huang & Kuo, 2002 [7] Huang, Kuo, & Lyu (2007)[9],Kapur, Gupta, Shatnawi, & Yadavalli (2006)[13]
$N \frac{(mt)^b}{1+(mt)^b}$	Log-Logistic	Gokhale & Trivedi, (1999)[5], Bokhari & Ahmad,(2006)[3]
$N(1 - e^{-\alpha t})^b$	Generalized Exponential	Quadri, Ahmad, Peer, & Kumar, (2006)[22]
$N(1 - e^{-mt^\delta})^b$	Exponentiated Weibull	Ahmad, Khan, & Rafi,(2010)[2]
$N \frac{1-e^{-\alpha t}}{1+Ae^{-\alpha t}}$	Inflexion S-Shaped	Q. Li, Li, Lu, & Wang, (2011)[14]

Table 1: Testing-Effort Functions proposed in Literature

2.3. **FRF**

Musa (1975) coined the term FRF and defined it as “ the proportion of failures experienced to the number of faults removed” (Musa [18]). Several others defined FRF differently. Malaiya, Von Mayrhauser, & Srimani, (1993)defined FRF in terms of fault exposure ratio (Malaiya et al. [17]). Friedman, Tran, & Goddard, (1995)defined FRF in terms of three ratios namely, detectability, associability and fault growth[4]. Hsu, Huang, & Chang,(2011)discussed constant, increasing and decreasing curves of FRF and validated these trends on six real life fault dataset (Chang [6]).Pachauri et al.(2015) considered Inflexion S-Shaped FRF to increase accuracy of growth model (Dhar et al. [21]). Later Aggarwal, Dhaka, & Nijhawan(2017) proposed Exponentiated Weibull(EW) FRF based SRGM with change points (Aggarwal et al. [1]).

3. PROPOSED MODELLING FRAMEWORK

3.1. **Notations:**

The notations used for developing the proposed model framework are given as:

$W(t)$	Cumulative testing effort by time t
$m(W_t)$	Mean value function corresponding to testing effort $W(t)$ or the expected number of faults detected or removed by time t using cumulative testing effort $W(t)$.
a	Initial fault content of the software
r	Fault detection rate (FDR)
$B(W)$	Testing-effort dependent fault reduction factor
b_1	Scale parameter before Change point
b_2	Scale parameter after Change point
r_1	FDR before Change point
r_2	FDR after Change point
l_1	Shape parameter before Change point
l_2	Shape parameter after Change point

3.2. **Assumptions of the proposed model**

1. The process of fault detection /removal follow the Non-homogeneous Poisson process (NHPP).
2. The software systems are subject to random failures caused by the faults remaining in the system.
3. All the faults are mutually independent and failure rate is uniformly affected by all the faults remaining in the software.
4. The expected number of faults detected/removed in time interval of length dt is proportional to the number of faults remaining in the system.
5. Fault detection /removal rate may change at any time moment known as change point.

6. The factor of proportionality is a function of time –dependent FRF modelled by logistic function.
7. The testing effort is modelled by Weibull-function.

3.3. *A brief description of Non- Homogenous Poisson Process*

A Counting process $\{N(t) ; t \geq 0\}$ is said to be Non-Homogeneous Poisson Process (NHPP) if it satisfies the following assumptions:

1. $N(0)=0$
2. $P(\text{exactly 1 failure in time interval}(t,t+\Delta t))=\lambda(t)\Delta t+o(\Delta t)$
3. $P(\text{more than 1 failures in time interval}(t,t+ \Delta t))= o(\Delta t)$
4. Counting process has independent increments.

Here, Expected value of $N(t)=m(t)$, i.e. Mean Value Function. NHPP based SRGM are either concave or S-shaped depending upon the shape of failure curve described by them.

3.4. *Model Development*

The rate of change of mean value function $m(W)$ can be represented by the following differential equation:

$$\frac{d}{dt}m(W) = r(W) (a - m(W)) \frac{dW}{dt} \quad (1)$$

$$\text{and } r(W) = r \times B(W) \quad (2)$$

where $r(W)$ and $B(W)$ are the testing effort dependent fault detection rate and FRF respectively.

3.5. *Weibull Testing Effort Function (TEF)*

In this paper, the cumulative testing effort consumption W_t is modeled by Weibull distribution. The Weibull TEF is flexible and can fit various types of resource consumption data and has been used in many applications for solving multi-disciplinary problems. It very well models initial increase and eventual decline in resource consumption behavior.

The Weibull TEF is given as:

$$W(t) = \bar{W} \times (1 - e^{-\alpha t^k}) \quad (3)$$

where,

\bar{W} :Total available testing effort

k, α : Shape and scale parameters of Weibull TEF ; $\alpha > 0$, $k > 0$

The corresponding instantaneous rate of effort consumption is given by:

$$\frac{d}{dt}W(t) = w(t) = \bar{W} \alpha k t^{k-1} e^{-\alpha t^k} \quad (4)$$

3.6. **Effort-dependent FRF**

Fault Reduction Factor has been modelled by logistic curve given as follows:

$$B(W) = \frac{l}{(1 + be^{-lW})} \tag{5}$$

Where, l and b denote the shape and scale parameter respectively of logistic function.

As the testing progresses with the learning of the testing team, the fault reduction factor initially increases with the increasing rate and after certain time, increases with the decreasing rate with the saturation of the learning of the testing team. Therefore, we have selected Logistic FRF to model this type of behavior.

Case1: Perfect debugging without change point.

The corresponding differential equation is:

$$\frac{d}{dW}m(W) = \frac{dm/dt}{dW/dt} = r(W) (a - m(W)) \tag{6}$$

$$\frac{d}{dW}m(W) = r \times B(W) (a - m(W)) \tag{7}$$

$$\frac{d}{dW}m(W) = r \times \frac{l}{(1 + be^{-lW})} (a - m(W)) \tag{8}$$

On integrating and using $m(0) = 0$ we get:

$$m(W) = a \left[1 - \frac{(1 + b)^r e^{-lrW}}{(1 + be^{-lW})^r} \right] \tag{9}$$

Case2: Perfect debugging with change point

The differential equation for this case can be written as:

$$\frac{\frac{d}{dt}m(W)}{\frac{dW}{dt}} = r(W) (a - m(W)) \tag{10}$$

Where $r(W) = r \times B(W)$

$$\text{and } r = \begin{cases} r_1, & t \leq \tau \\ r_2, & t > \tau \end{cases}$$

$$B(W) = \begin{cases} \frac{l_1}{1 + b_1 e^{-l_1 W}}, & t \leq \tau \\ \frac{l_2}{1 + b_2 e^{-l_2 W}}, & t > \tau \end{cases}$$

On integrating the above differential equation, we get:

$$m(W) = a \left[1 - \frac{(1 + b_1)^{r_1} e^{-l_1 r_1 W}}{(1 + b_1 e^{-l_1 W})^{r_1}} \right], t \leq \tau \tag{11}$$

$$m(W) = a \left[1 - \frac{(1 + b_1)^{r_1} e^{-l_1 r_1 W(\tau) - l_2 r_2 (W - W(\tau)) \left(\frac{1 + b_2 e^{-l_2 W(\tau)}}{1 + b_2 e^{-l_2 W}} \right)^{r_2}}}{(1 + b_1 e^{-l_1 W(\tau)})^{r_1}} \right], t > \tau \tag{12}$$

Dataset	Description	Testing time (Weeks)	Execution time (CPU hours)	Faults	Change Point (Weeks)
DS-1	Tandem Computers	20	10000	100	8
DS-2	PL/I database application software system	19	47.65	328	6

Table 2: Datasets

Parameters	DS-1	DS-2
W	11710.754	799.016
α	0.024	0.002
K	1.46	1.115

Table 3: Estimated parameters of Weibull TEF for two datasets

4. NUMERICAL EXAMPLE

To determine the accuracy of the proposed model, we validate it on two fault datasets. Data set 1 (DS-1) was given by Obha & Yamada, (1984) and second dataset (DS-2) was reported by Wood, (1996) (Obha and Yamanda [20]; Wood [23]).

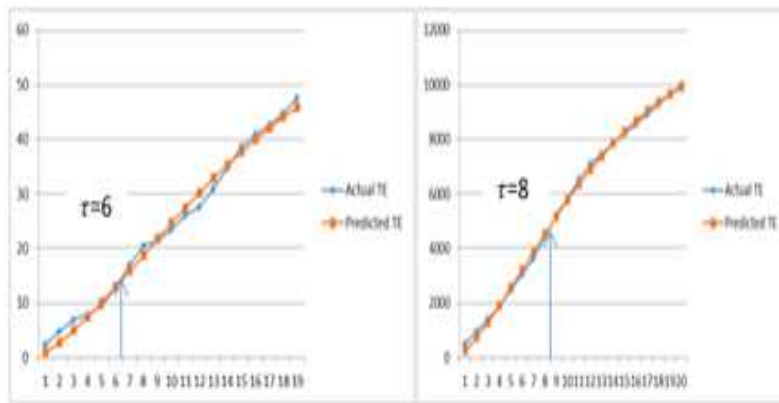
The first data set (DS-1) had been collected during 19 weeks of testing a real time command and control system of size 1317 KLOC and 328 faults were detected during testing. The second data set (DS-2) had been collected from report of tandem computers where the testing was done for 20 weeks and 100 faults were detected during the testing phase. To identify the change points, we plotted the curves corresponding to the actual failure dataset and observed the position of kinks (sudden change in the shape of curve) on the curve. The time-point corresponding to the kink is known as change point. The change points for the data sets under consideration have been observed at the end of 6 weeks (for DS-1) and 8 weeks (for DS-2) respectively.

Datasets are described in Table 2 below.

For both the datasets, we predict the consumption corresponding to Weibull TEF. The estimated values of Weibull TEF parameters obtained for both datasets are given in Table 3.

We focus on data related to testing-effort which is given by execution time in

CPU hours per week. Figure 1 shows the comparison between actual and predicted consumption using Weibull TEF curve.



(a)DS-1 (b)DS-2

Figure 1: Goodness of fit curves for testing effort corresponding to two datasets

Based on the predicted TEF, we further use LSE technique to estimate the faults. The estimation results for the proposed model are shown in Table 4. Figures 2 and 3 depict the curves of predicted and actual faults for the two datasets respectively.

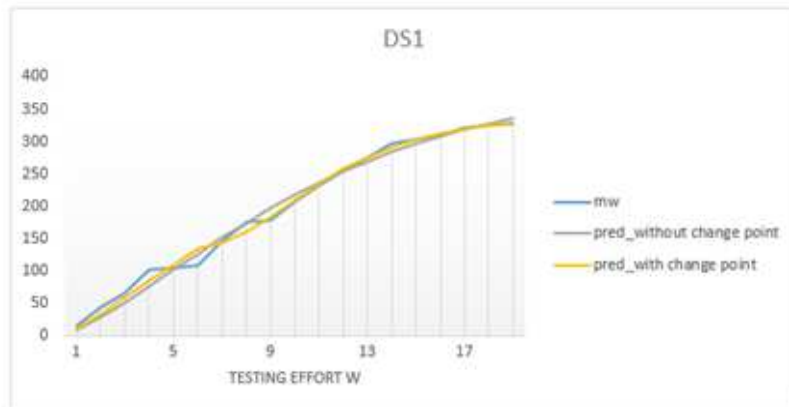


Figure 2.: Goodness of fit curve for DS1

Model-1 Parameters	DS-1	DS-2
a	110.751	567.473
b	0.945	0.055
r	0.005	0.044
l	0.043	0.443
Model-2 Parameters	DS-1	DS-2
a	108.947	353.656
b_1	0.044	0.003
b_2	0.770	137.64
r_1	0.004	0.071
r_2	0.032	0.373
l_1	0.042	0.518
l_2	0.010	0.238

Table 4: Estimated Parameters for the two models

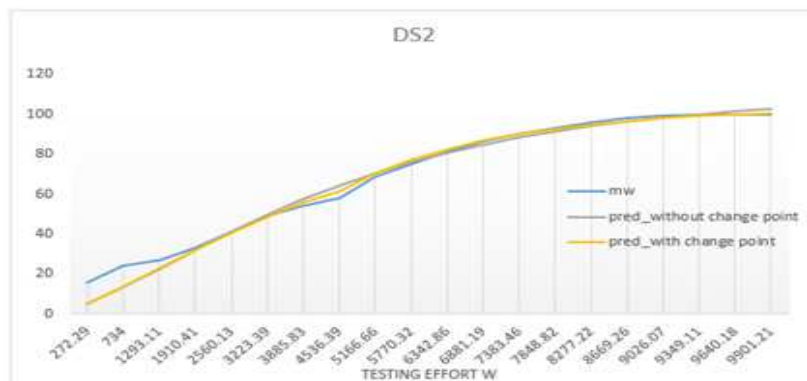


Figure 3: Goodness of fit curve for DS2

To demonstrate the accuracy of our proposed model for fitting the failure data set, we have worked out six Goodness of fit criteria values. The rationale behind using more than 1 Goodness of fit criteria is that we want to check whether accuracy of the model change if a different goodness of fit criteria is used. Six comparison criteria's by Lee et. al., (2018) have been used in our study. These are listed in Figure 4. For all the mentioned criteria except the first one, lower values signify better fit hence better accuracy of the model.

Dataset	Model	R^2	MSE	PP	PRR	PRV	RMSPE
DS-1	Model -1	0.965	16.58	13.92	33.26	4.06	4.17
	Model -2	0.984	13.39	12.63	30.53	3.62	3.75
DS-2	Model -1	0.988	123.56	23.04	28.95	11.28	11.41
	Model -2	0.993	73.17	13.71	14.42	8.72	8.78

Table 5: Model Performance Comparison

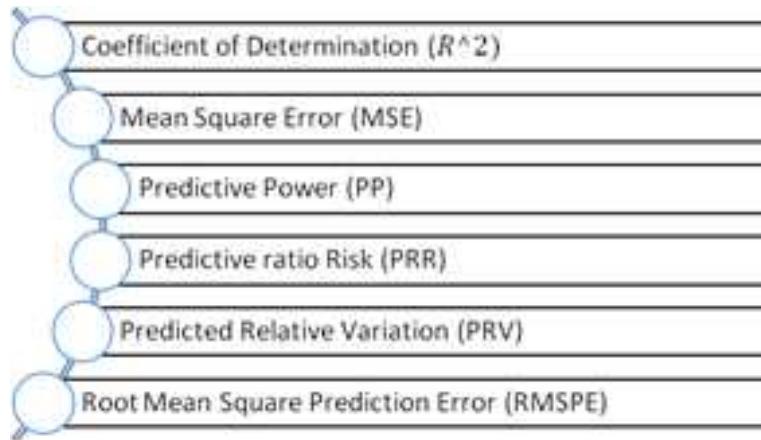


Figure 4: Performance Criteria

The values of performance criteria are based on the predicted faults for each of the dataset corresponding to the both models, presented in Table 5. We can observe that for all criteria, model considering change point along with FRF and testing effort gives better fitting.

Further, the comparison is done with the Kapur and Garg model using Weibull testing effort function Kapur, Gupta, Shatnawi and Yadavalli(2006), Hsu et al. (2011)model with constant FRF, and Pachauri et al. (2015)with Inflexion S-Shaped FRF(Gupta et al. [13]; Chang et al. [6]; Dhar et al. [21]). The comparison results are given below in Table 6.Also the corresponding Goodness of Fit curves are shown in Fig. 5.

Dataset	Model	R^2	MSE	PP	PRR	PRV	RMSPE
DS-1	Model -1	0.965	16.58	13.92	33.26	4.06	4.17
	Model -2	0.984	13.39	12.63	30.53	3.62	3.75
	Kapur,Gupta,Shatnawi and Yadavalli(2006) [13]	0.95	36.604	26.93	81.91	5.98	6.20
	Hsu model(2011) [6]	0.964	29.14	15.68	25.32	5.54	5.54
	Pachauri model(2015) [21]	0.965	28.64	18.03	39.64	5.47	5.49
DS-2	Model -1	0.988	123.56	23.04	28.95	11.28	11.41
	Model -2	0.993	73.17	13.71	14.42	8.72	8.78
	Kapur,Gupta,Shatnawi and Yadavalli(2006) [13]	0.984	169.07	33.05	45.74	12.99	13.34
	Hsu model(2011) [6]	0.985	157.76	27.25	24.03	12.88	12.90
	Pachauri model(2015) [21]	0.986	146.6	22.27	23.56	12.43	12.44

Table 6: Comparison with other model in Literature

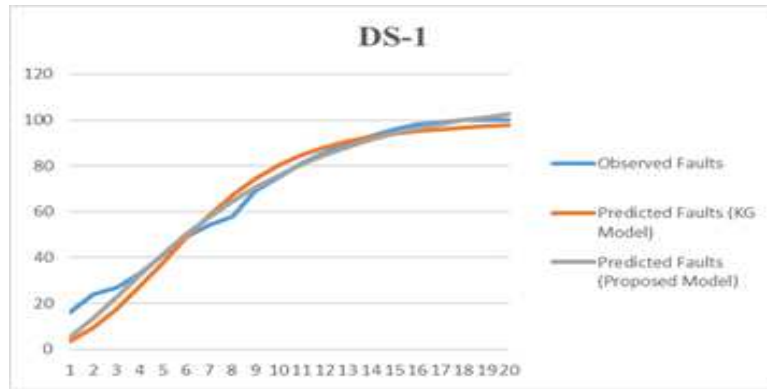


Figure 5(a) K-G model(DS-1)

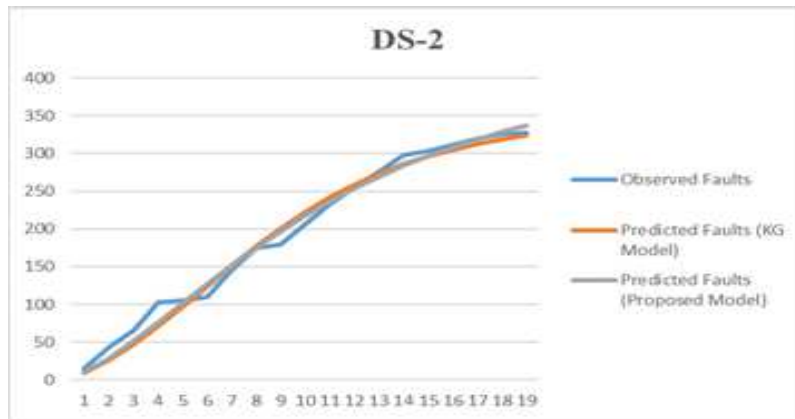


Figure 5(b) K-G model(DS-2)

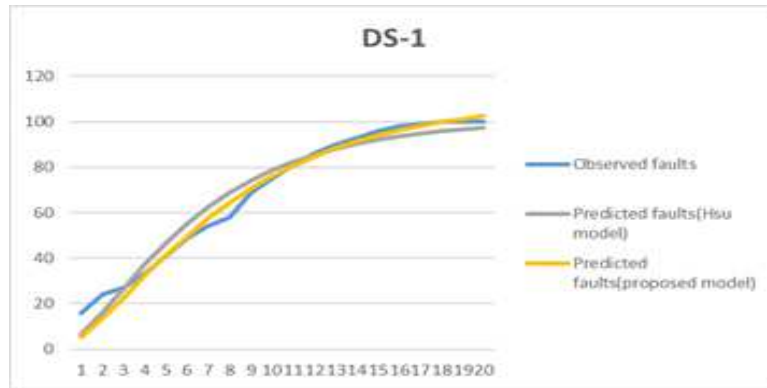


Figure 5 (c) Hsu model(DS-1)

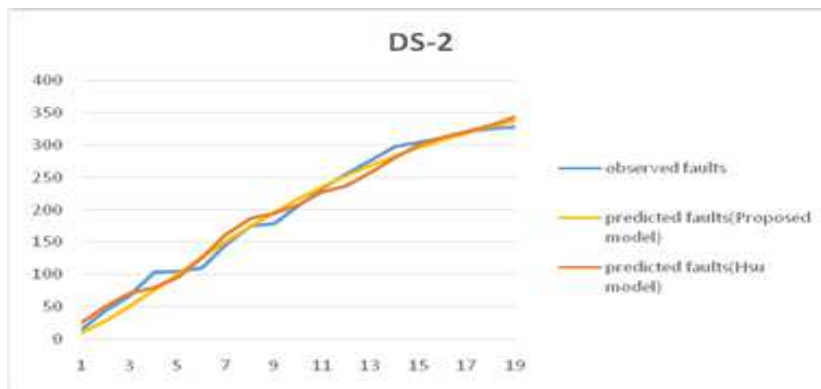


Figure 5 (d) Hsu model(DS-2)

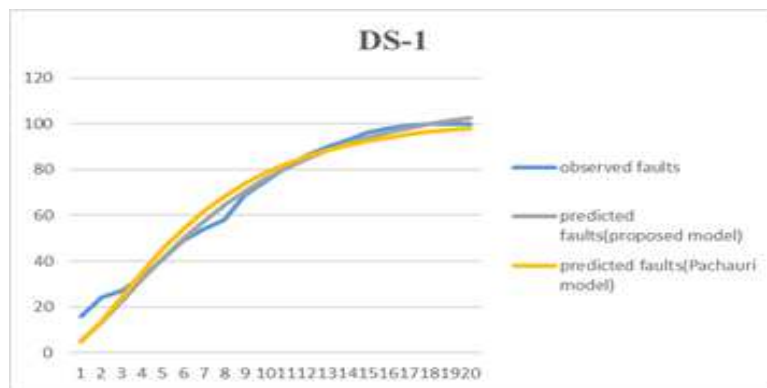


Figure 5 (e) Pachauri model(DS-1)

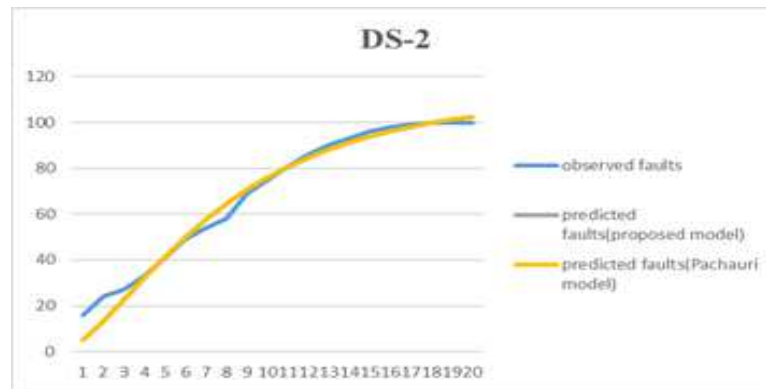


Figure 5(f) Pachauri model(DS-2)

Figure 5: Comparison using Goodness of Fit curves corresponding to two datasets

5. CONCLUSION

Here, we proposed two SRGMs based on testing effort dependent FRF and Change point. First model considers that there is no change in the FDR. While in the second model, we assume that at a particular time moment there is a change in the FDR, i.e., there is a change point. The testing effort function, has been modeled by Weibull function and FRF is represented by the logistic function. The models are tested on two real fault datasets obtained from literature. Also, a comparison is made with the models proposed by Kapur, Gupta, Shatnawi and Yadavalli(2006), Hsu, Huang, & Chang (2011) and Pachauri (2015) (Chang et al. [6]; Gupta et al. [13]; Dhar et al. [21]).The Least Square Estimation results from SPSS show that the model incorporating change point yield better results than the model without change point. Also, the comparison results with existing models in literature illustrate that the proposed model has better performance and is more accurate in predicting faults. The goodness of fit curves shows the closeness of actual and predicted values for both the effort function and the proposed model.

This model can be extended to incorporate multiple change points, imperfect debugging conditioned multi-release software growth modelling. Many a times due to fault dependency more errors get generated during the debugging activity, which also plays important role in the failure process. The model can be used to deal with the Optimal Release problems.

REFERENCES

- [1] Aggarwal, A. G., Dhaka, V., & Nijhawan, N., "Reliability analysis for multi-release open-source software systems with change point and exponentiated Weibull fault reduction factor", *Life Cycle Reliability and Safety Engineering*, 6(1)(2017) 3-14.
- [2] Ahmad, N., Khan, M. G., & Rafi, L. S., "A study of testing-effort dependent inflection S-shaped software reliability growth models with imperfect debugging", *International Journal of Quality & Reliability Management*, 27(1) (2010) 89-110.
- [3] Ahmad, N & Bokhari, M., Analysis of a software reliability growth models: the case of log-logistic test-effort function, *Paper presented at the Proceedings of the 17th IASTED international conference on modeling and simulation (MS'2006)*, Montreal, Canada.
- [4] Friedman, M. A., Goddard, P. I., & Tran, P. Y., *Reliability of software intensive systems*, William Andrew, Norwick, USA, 1995.
- [5] Gokhale, S. S., and Trivedi, K. S., "A time/structure based software reliability model", *Annals of Software Engineering*, 8 (1-4) (1999) 85-121.
- [6] Chang, J.-R., Hsu, C.-J., & Huang, C.-Y., "Enhancing software reliability modeling and prediction through the introduction of time-variable fault reduction factor", *Applied Mathematical Modelling*, 35(1) (2011) 506-521.
- [7] Huang, C.-Y., and Kuo, S.-Y., "Analysis of incorporating logistic testing-effort function into software reliability modeling", *IEEE Transactions on Reliability*, 51(3) (2002)261-270.
- [8] Chen, Y., Huang, C.-Y., & Kuo, S.-Y., "Analysis of a software reliability growth model with logistic testing-effort function", *Paper presented at the Proceedings The Eighth International Symposium on Software Reliability Engineering*, Albuquerque, USA, 1997.
- [9] Huang, C.-Y., Kuo, S.-Y., & Lyu, M. R., "An assessment of testing-effort dependent software reliability growth models", *IEEE Transactions on Reliability*, 56(2) (2007) 198-211.
- [10] Huang, C.-Y., Kuo, S.-Y., & Lyu, M. R., "A unified scheme of some nonhomogenous poisson process models for software reliability estimation", *IEEE transactions on software engineering*, 29(3) (2003) 261-269.
- [11] Inoue, S., and Yamada, S., "Software reliability measurement with effect of change-point: Modeling and application", *International Journal of System Assurance Engineering and Management*, 2(2) (2011) 155-162.
- [12] Goswami, D., Gupta, A., & Kapur, P., "A software reliability growth model with testing effort dependent learning function for distributed systems", *International Journal of Reliability, Quality and Safety Engineering*, 11(4) (2004) 365-377.
- [13] Gupta, A., Kapur, P., Shatnawi, O., & Yadavalli, V., "Testing effort control using flexible software reliability growth model with change point", *International Journal of Performance Engineering*, 2(3) (2006) 245-262.
- [14] Li, H., Li, Q., Lu, M., & Wang, X., "Software reliability growth model with S-shaped testing effort function", *Journal of Beijing University of Aeronautics and Astronautics*, 37(2), (2011).
- [15] Li, X., Ng, S. H., & Xie, M., "Sensitivity analysis of release time of software reliability models incorporating testing effort with multiple change-points", *Applied Mathematical Modelling*, 34(11)(2010) 3560-3570.
- [16] Huang, C.Y. and Lin, C.T., Software Reliability Modeling with Weibull-type Testing-Effort and Multiple Change-Points, *Paper presented at the TENCON 2005-2005 IEEE Region 10 Conference*, La Trobe University, Melbourne, 2005.
- [17] Malaiya, Y. K., Srimani, P. K., & Von Mayrhauser, A., "An examination of fault exposure ratio", *IEEE transactions on software engineering*, 19(11)(1993) 1087-1094.
- [18] Musa, J. D., "A theory of software reliability and its application", *IEEE transactions on software engineering*, (3)(1975) 312-327.
- [19] Musa, J. D., Lannino, A., Okumoto, K., "Software Reliability, Measurement", *Prediction, Application, McGraw-Hill*, 61 (1987) 46-51.
- [20] Obha, M., and Yamada, S., "S-shaped software Reliability Growth Mode", *Paper presented at the Proceedings of the 4th National Conference on Reliability and Maintainability*, San Francisco, CA, USA, (1984) 475-484.
- [21] Dhar, J., Kumar, A., Pachauri, B., "Incorporating inflection S-shaped fault reduction factor

- to enhance software reliability growth”, *Applied Mathematical Modelling*, 39 (5-6)(2015) 1463-1469.
- [22] Ahmad, N., Kumar, M., Peer, M., & Quadri, S., “Nonhomogeneous Poisson process software reliability growth model with generalized exponential testing effort function”, *RAU Journal of Research*, 16 (1-2) (2006) 159-163.
- [23] Wood, A., “Software reliability growth models”, *Tandem technical report*, 96 (130056)(1996).
- [24] Hishitani, J., Osaki, S., & Yamada, S., “Software-reliability growth with a Weibull test-effort: a model and application”, *IEEE Transactions on Reliability*, 42(1) (1993) 100-106.
- [25] Wang, J. & Zhao, J., “Testing the existence of change-point in NHPP software reliability models”, *Communications in Statistics—Simulation and Computation*, 36(3)(2007) 607-619.
- [26] Zou, F. Z., “A change-point perspective on the software failure process”, *Software testing, verification and reliability*, 13(2)(2003) 85-93.