Global Journal of Modeling and Intelligent Computing (GJMIC)

Vol. 3, No. 1, January, 2023 **ISSN: 2767-1917**

A GENERALIZED RELIABILITY GROWTH MODEL FOR SOFTWARE SYSTEM OPERATING IN RANDOM ENVIRONMENT

P. Agarwal, SRM Institute of Science & Technology, India (priyanka1354@gmail.com)
M. Jain, Indian Institute of Technology Roorkee, India(madhu.jain@ma.iitr.ac.in)
S. Kumar, Indian Institute of Technology Roorkee, India(skumar@ma.iitr.ac.in)

ABSTRACT

In the current era of modern technology, a human cannot think to survive without software as such key area of attention of software manufacturer is to produce bug-free software and maintain the reliability and compatibility with human activities dependent upon software embedded devices. The manner in which the software will perform in a random field environment is a very major issue to study. Taking effects of random field environment into account, in this paper we develop a generalized software reliability growth model (G-SRGM) with generalized fault coverage function. For demonstrating the better performance of the proposed model, two data sets are taken and computational results of proposed models are compared with the existing models using Least Square Estimation (LSE) technique in MATLAB software. The three goodness-of-fit criteria such as the sum of square error, R-square, and root mean square error are also used for comparison.

Keywords: Software Reliability, Non-Homogeneous Poisson Process, Testing Coverage, Random Operating Environment, Least Square Estimation.

1. INTRODUCTION

With the progression of computer technology, human life is being entirely dependent on software embedded into gadgets/equipment, manufacturing/assembly lines, and many more systems. The development of reliable and sustainable software is a very daunting job for software manufactures. Many software projects cannot afford the failure of software because in many cases it affects directly or indirectly human life such as in Medical Appliances, Military weapons, Safety-critical systems, etc. Several non-homogeneous Poisson process (NHPP) based software reliability growth models (SRGMs) have been proposed in literature but no generalized model is available that can predict the mean value function, reliability and optimization issues and other parameters of model with generalized fault coverage function in the scenario when software operates in random environment.

Many software reliability growth models have been discussed in the last three decades and still, immense researches are also in the process to achieve the pre-specified reliability of the real time software system. The most classic model to analyze software failures was developed by Goel and Okumoto (1979). After that, Obha (1984) and Obha et al. (1984) extended the work presented in the article of Goel&Okumoto (1979) by considering the different failure rates. A plenty number of research papers have been available in SRGM literature having several different conditions such as perfect/imperfect debugging, various types of testing effort functions (TEF's), coverage factor, change-point phenomenon, delay function, environment effects and many more.

Software reliability growth models are widely used to facilitate the more precise information regarding failures and reliability according to the user's need. In 1996, Gokhale et al. proposed the enhanced non-homogeneous Poisson process (NHPP) based model by incorporating the test coverage functions during the testing and operational both phases. Pham and Zhang (2003) used the testing coverage function in the analytical model based on NHPP to discuss the failure/faults and obtained the expected cost and optimal release policies. Xie and Yang (2003) analyzed software model by including the imperfect debugging feature which is a realistic aspect in the operating environment in particular when the imperfect debugging affects the software development cost. Chatterjee and Shukla (2016) examined the effects of test coverage function along with change-point and time-dependent fault detection to study the software reliability. Rani et al. (2021) proposed a software reliability model considering the uncertainties of field environments using testing coverage function. Recently, Yadav and Yadav (2023) used machine learning methods to optimize and predict the software reliability.

During last two decades, some models have appeared which have included the assumptions of random operating environment and developed SRGMs by using the NHPP. Zhang et al. (2002) discussed the dissimilarity between the system testing and operating environment. Zhang and Pham (2006) used a field failure rate to balance the testing and field reliability, cost and other parameters for making a better decision. Teng and Pham (2006) proposed a new methodology subject to a random field environment. Chang et al. (2014) and Pham (2014) discussed the SRGMs subject to random field environment with loglog fault detection rate and testing coverage function. Again Pham (2014) used a Vtub-shaped fault detection rate to predict better results in the field environment. Zhu et al. (In 2015) surveyed the various environment functions and their impact on software reliability assessment incorporating different categories. They have used various statistical methods including the goodness-of-fit criteria to analyze the software's behavior in the operating field. Zhu and Pham (2016) and Li and Pham (2017) extended the theory of randomness of field environment using testing coverage function in the more realistic situation by incorporating the imperfect debugging. Song et al. (2019) analyzed the sensitivity of reliability function using new testing coverage function under random operating environment. Li and Pham (2019) covered the imperfect debugging phenomenon to explore the SRGM in uncertain environment. Recently, Zhu and Pham (2020) suggested a new reliability model in multiple environments to examine the effects with stochastic fault detection process. Li and Pham (2021) discussed he fault detection and correction process using different fault amount. Lin and Huang (2022) used queueing based simulation for analyzing reliability.

In this paper, we consider a generalized S-shaped testing coverage function to develop the SRGM. The NHPP based formulism is done to demonstrate the effects of random operating environment. For specific cases of our model, we compare the obtained results with the existing ones. The generalized proposed model affirms the better outcomes by using the goodness of fit criteria than existing ones. For this reliability indices prediction purpose, the expected number of detected faults and reliability using the generalized testing coverage function and random operating environment are derived in section 2. Section 3 is devoted to the specific testing coverage function and their respective mean value function. Software failure data is given in section 4 and the proposed model is compared with the existing models in this section based on goodness-of-fit criteria. Finally, concluding remarks are given in section 5.

2. MODEL DESCRIPTION

In this paper, we develop SRGM to analyze the effects of generalized testing coverage function in random operating environment and present the analytical formulation for the expected number of detected faults and reliability. For the mathematical formulation of SRGM, we assume that the mean number of failures per unit time follows the non-homogeneous Poisson process (NHPP). The expected number of detected faults in the random field environment is measured by the following differential equation (Pham, 2016)

$$\frac{dm(t)}{dt} = \eta C_k(t)\left[A - m(t)\right] \quad \dots (1)$$

where A is the fault content function and η denotes the uncertainty of the testing coverage function in the random field environment with a generalized probability density function with two parameters $\alpha,\beta \geq 0$. So that mean value function m(t) in the general form (G-SRGM) can be obtained from equation (1)

$$m(t) = A\left[1 - \frac{\beta}{\beta + \int_0^t C_k(s)\, ds}\right]^{\alpha} \quad (2)$$

2.1 Testing Coverage

Testing coverage is the keynote feature of software testing to figure out how many faults have been covered and how much effort is required to fulfill the pre requisitions of the customer's interest. We denote the the parameter reflecting the quality of testing by δ .Here we use the following generalized testing coverage function (G-TCF).

$$C_k(t) = \frac{c_k^{'}(t)}{1-c_k(t)} \tag{3}$$

where $c_k(t) = [1 - R(k,t)e^{-\delta t}]$ and $R(k,t) = \sum_{i=0}^{k} \frac{(\delta t)^i}{i!}$. (4)

3. PERFORMANCE INDICES

3.1 Mean value function

Using equations (3) and (4) in equation (2), we get the G-SRGM as

$$m(t) = A\left[1 - \frac{\beta}{\beta + \delta t - \ln\left\{k!\sum_{i=0}^{k-1}\frac{(\delta t)^i}{i} + (\delta t)^k\right\} + \ln k!}\right]^\alpha \quad \ldots(5)$$

3.2 Reliability Evaluation

By assuming that the software faults will not occur in operational phase (T, T+x] and ($T \geq 0$, $x > 0$), the reliability function for the software can be obtained as

$$R(x/T) = \exp\left[-\left\{m(T+x) - m(T)\right\}\right] \quad \ldots(6)$$

4. SPECIFIC TESTING COVERAGE FUNCTIONS AND MEAN VALUE FUNCTIONS

For the validity of the proposed model, in this section we are considering the different values of k and analyze the effects of coverage function on the mean value function. It is evident that as the value of k increases, more faults can be covered The three cases are considered:

Case (i): When k=1

$$c_1(t) = 1 - (1 + \delta t)e^{-\delta t} \tag{7a}$$

$$m(t) = A\left[1 - \frac{\beta}{\beta + \delta t - \ln(1 + \delta t)}\right]^\alpha \tag{7b}$$

Case (ii): When k=2.

$$c_2(t) = 1 - (1 + \delta t + \frac{\delta^2 t^2}{2!})e^{-\delta t} \tag{8a}$$

$$m(t) = A\left[1 - \frac{\beta}{\beta + \delta t - \ln(2 + 2\delta t + \delta^2 t^2) + \ln 2!}\right]^\alpha \tag{8b}$$

Case (iii): When k=3

$$c_3(t) = 1 - (1 + \delta t + \frac{\delta^2 t^2}{2!} + \frac{\delta^3 t^3}{3!})e^{-\delta t} \tag{9a}$$

$$m(t) = A\left[1 - \frac{\beta}{\beta + \delta t - \ln\left(6 + 6\delta t + 3\delta^2 t^2 + \delta^3 t^3\right) + \ln 3!}\right]^{\alpha} \tag{9b}$$

The mean vale functions for some well known models and proposed generalized model are summarized in table 1.

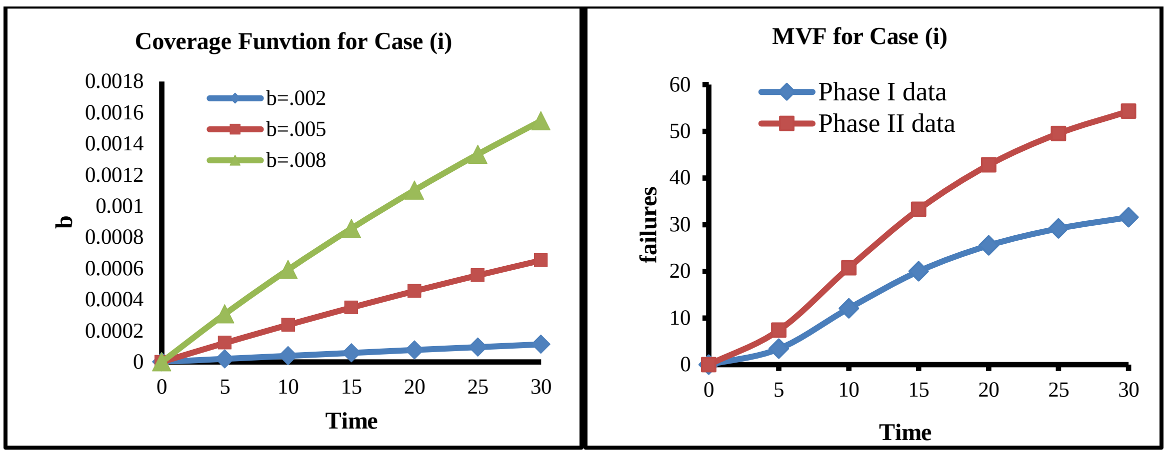
Table 1: Mean value functions for the existing and proposed NHPP based SRGM.

S. No.	Model name	Researchers	Mean Value Function [m(t)]
1.	Goel-Okumoto (G-O) Model	Goel and Okumoto (1979)	$m(t) = a(1 - e^{-bt})$
2.	Delayed S-Shaped Model	Obha and Yamada (1984)	$m(t) = a(1 - (1 + bt)e^{-bt})$
3.	Inflection S-Shaped Model	Obha (1984)	$m(t) = \frac{a(1 - e^{-bt})}{1 + \beta e^{-bt}}$
4.	Yamada imperfect debugging Model	Pham (2014)	$m(t) = a[1 - e^{-bt}][1 - \frac{\alpha}{b}] + \alpha a t$
5.	PNZ Model	Pham et al. (1999)	$m(t) = \frac{a[1 - e^{-bt}][1 - \frac{\alpha}{b}] + \alpha t}{1 + \beta e^{-bt}}$
7.	V tub-shaped fault detection rate model	Pham, H. (2014)	$m(t) = A\left[1 - \left(\frac{\alpha}{\alpha + a^{t^b} - 1}\right)^n\right]$
8.	Chang's testing coverage model	Chang et al. (2014)	$m(t) = A\left[1 - \left(\frac{\alpha}{\alpha + at}\right)^n\right]$
9	Logistic fault detection model	Pham, H. (2016)	$m(t) = A\left[1 - \frac{\alpha}{\alpha + \frac{c}{b}\ln\left(\frac{a + e^{bt}}{1 + a}\right)}\right]^{\eta}$
10	Generalized SRGM	Proposed model	$m(t) = A\left[1 - \frac{\beta}{\beta + \delta t - \ln\left\{k!\sum_{i=0}^{k-1}\frac{(\delta t)^i}{i} + (\delta t)^k\right\} + \ln k!}\right]^{\alpha}$

5. SOFTWARE FAILURE DATA AND NUMERICAL RESULTS

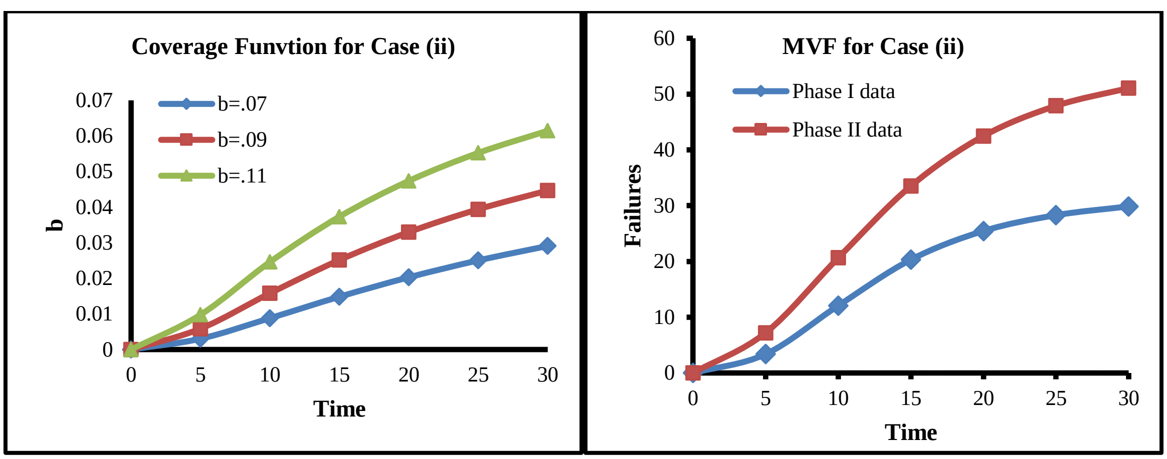
To validate the findings of the suggested model, test data of telecommunication system for two phases (phase I and Phase II) given by Zhu and Pham (2016) are used and comparison of our model with the existing models are presented in Tables 2 and 3, respectively. For each phase, the number of detected faults during each week of the test is recorded and the aggregate number of faults is observed. As shown in Tables 2 and 3, the system test hours per week are observed as 356 and 416 for phase I and II data, respectively. To perform better results and discussion, parameter estimation was implemented using Least Square Estimation (LSE) in MATLAB Software.

From figs 1(a), 2(a) and 3 (a), we interpret the testing coverage function by varying the parameter δ. It is seen that as δ is increasing, the testing coverage function is also increasing rapidly and their respective mean value functions are also figured out for both phases of data. Figs 4(a & b) depict the comparison of the mean value function of the proposed model with the existing models for both phases of data I and II, respectively. It is observed that most of the faults are covered for both phases.



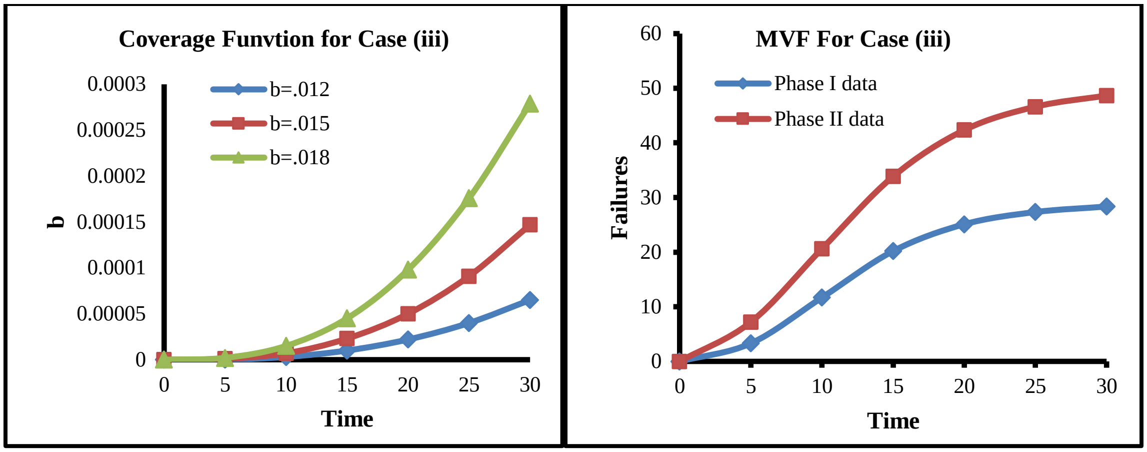
(a) (b)

Figure 1: (a) Testing Coverage Function by varying b and (b) MVF for case (i)



(a) (b)

Figure 2: (a) Testing Coverage Function by varying b and (b) MVF for case (ii)



(a) (b)

Figure 3:(a) Testing Coverage Function by varying b and (b) MVF for case (iii)

Tables 4 & 5 summarize the results of the estimated parameters and the goodness-of-fit criteria such as Sum of Square Error (SSE), R-Square and Root Mean Square Error (RMSE). The statistical indices of all 9 models for both data sets given in table 1, are computed using the Least Square Estimation (LSE) technique in MATLAB software. For the illustration purpose, we also evaluate the mean value function for k=1, 2, 3 as case 1, case 2, and case 3, respectively. It is observed from tables 4 and 5 that on increasing the value of k, the number of errors decreases i.e., as a result the efficiency and compatibility of the software improve. In the proposed model, SSE=23.27 & RMSE=1.17 (for case 1), SSE=19.71 & RMSE=1.077 (for case 2) and SSE=16.83 & RMSE=0.9948 (for case 3) for phase I data are noticed. The computational results of SSE& RMSE for both cases are recorded in Table 5 by considering the phase II data.

Table 4: Model parameter estimation and comparison criteria for phase I data set

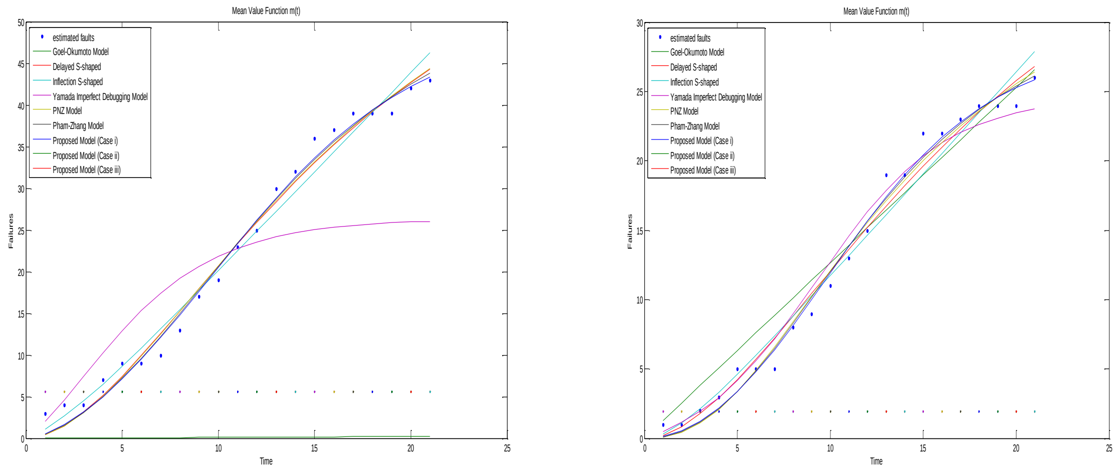
Model		**a**	**b**	**δ**	**c**	**n**	**A**	**α**	**β**	**SSE**	**R-Square**	**RMSE**
Goel-Okumoto (G-O)		10790	0.0001174	-	-	-	-	-	-	73.3	0.9546	1.964
Delayed S-Shaped		39.82	0.1104	-	-	-	-	-	-	28.38	0.9824	1.222
Inflection S-Shaped		26.69	0.2919	-	-	-	-	-	21.71	12.14	0.9925	0.8212
Yamada imperfect debugging		0.02789	0.5418	-	-	-	-	51.91		43.55	0.973	1.555
PNZ		1.032e-10	0.388	-	-	-	-	1.317	5.762	32.87	0.9796	1.351
Pham-Zhang		0.004031	0.2919	-	26.68	-	-	0.2069	21.68	12.14	0.9925	0.8711
V tub-shaped fault detection rate		0.7784	0.0261	-	-	0.3981	495.7	0.934	-	76.16	0.9528	2.182
Chang et al. (2014)		0.04105	2.187	-	-	220.5	27.15	54.42	-	17.59	0.9891	1.049
Logistic fault detection		1.19	14.45	-	0.223	-	44.58	0.05575	49.49	46.41	0.9712	1.759
Proposed model	Case 1	-	-	0.00443	-	1.229	38.99	0.001526	-	23.27	0.9856	1.17
	Case 2	-	-	0.0155	-	0.7147	32.83	0.00169	-	19.71	0.9878	1.077
	Case 3	-	-	0.0156	-	0.4949	29.54	0.00012	-	16.83	0.9896	0.9948

Table 5: Model parameter estimation and comparison criteria for phase II data set

Model Name		a	b	δ	c	n	A	α	β	SSE	R-Square	RMSE
Goel-Okumoto (G-O) Model		10200	0.0002088	-	-	-	-	-	-	125.9	0.9694	2.574
Delayed S-Shaped Model		62.3	0.119	-	-	-	-	-	-	62.19	0.9849	1.809
Inflection S-Shaped Model		46.54	0.240	-	-	-	-	-	12.22	33.67	0.9918	1.368
Yamada imperfect debugging Model		1.985	0.415	-	-	-	-	1.2	-	89.45	0.9783	2.229
PNZ Model		1.86e-10	0.311	-	-	-	-	2.195	1.964	76.74	0.9814	2.065
Pham-Zhang Model		1.023	0.241	-	46.43	-	-	0.00762	12.16	33.7	0.9918	1.451
V tub-shaped fault detection rate model		0.9577	0.009712	-	-	4.795	1073	0.4854	-	129.7	0.9685	2.847
Chang et al. Model (2014)		0.05058	2.102	-	-	182.6	43.58	66.81	-	62.14	0.9849	1.971
Logistic fault detection model		5.608	31.13	-	0.118	73.17	74.9	0.01981	-	115.8	0.9719	2.778
Proposed model	Case 1	-	-	0.00341	-	0.8905	70.56	0.00168	-	59.33	0.9856	1.868
	Case 2	-	-	0.01037	-	0.5572	57.2	0.00090	-	51.4	0.9875	1.739
	Case 3	-	-	0.01502	-	0.4038	51.18	0.00016	-	45.57	0.9889	1.637

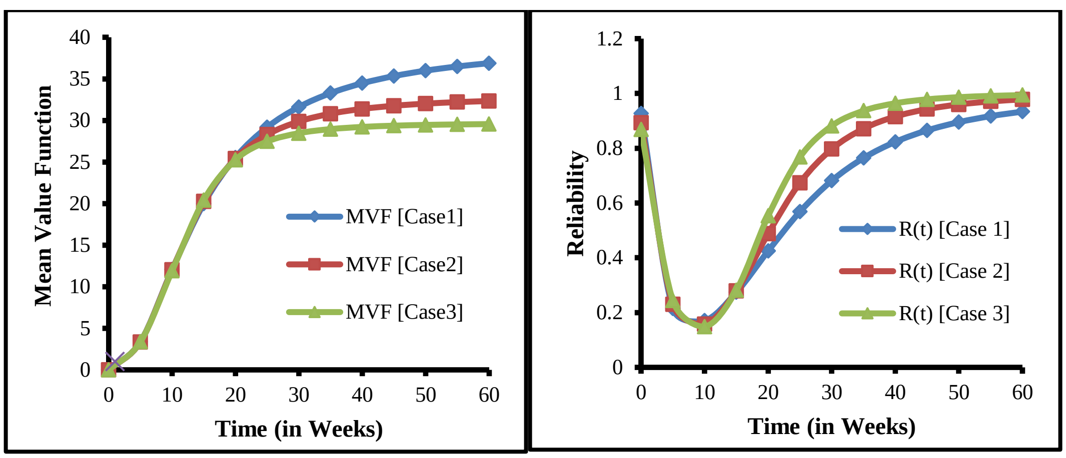
All goodness-of-fit (SSE, $R^{2,}$ and RMSE) values are less than the other existing models for phase II also. Only the inflection S-shaped model (Obha, 1984) is depicting the lower values than any other model so the developer can choose the model according to the user's prerequisite. Overall, it is noticed that SSE, R-Square and RMSE values are lower as such the proposed model is much better and compatible with random field environment in comparison to earlier model excluding inflection S-shaped model. The numerical illustration reveals that if the developer wants to improve the reliability of the software, it can possible by increasing the value of k in G-TCF according to the requirement of the user.

Figure. 5 (a) &5 (b) and Figure. 6(a) & 6(b) depict the trends of expected number of faults removed and their correspondence reliability for the real data set of Phase I and II, respectively. The mean value function and reliability are calculated for the n=1, 2 and 3 and maximum reliability is achieved for n =3. It is concluded that as the value of n increases, the reliability also increases. It can be concluded from the Figures that if the testing runs for 22 weeks and if we get the targeted reliability then the developer can release the software otherwise testing time can also increases.



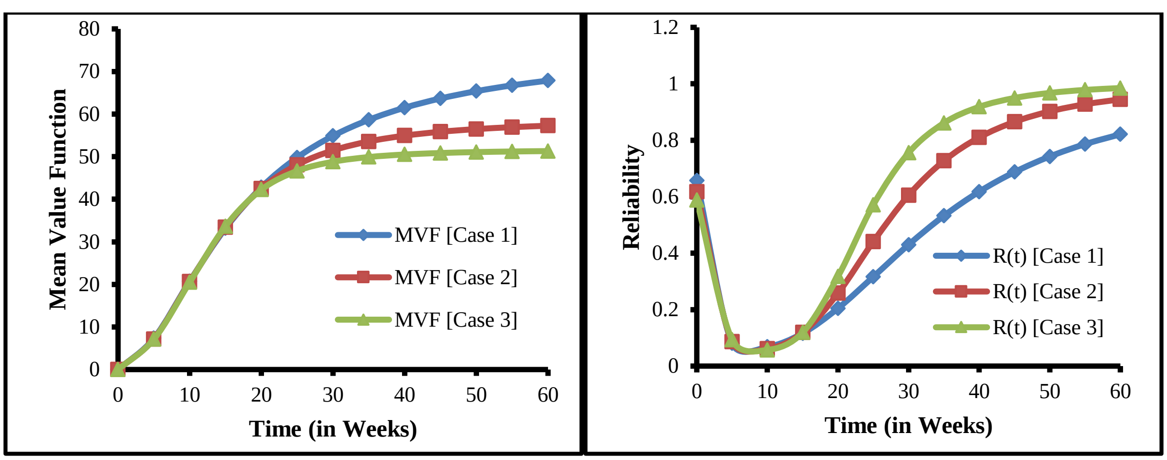
(a) **(b)**

Figure 4: Comparison of proposed Mean Value Function with existing models for (a) phase I and (b) Phase II data sets



(a) **(b)**

Figure 5: Mean Value Function and Reliability for Case 1, 2 and 3 for the Phase I Data



(a) **(b)**

Figure 6: Mean Value Function and Reliability for Case 1, 2 and 3 for the Phase II Data

6. CONCLUSIONS

Keeping in mind the dependency of human life on the software operating in random field environment such as its testing environment, we have proposed a generalized software reliability growth model (G-SRGM) by including the more realistic feature of generalized testing coverage function (G-TCF). For interpretation purpose, G-TCF was considered for different level of testing efficiency (i.e. k=1,2,3).It was demonstrated that as the testing efficiency increases, more faults are covered. For the validation of the purpose, real time telecommunication system test data for two phases are considered. It is validated by taking numerical results that the proposed model is having the minimum errors during the estimation and on increasing the value of k in G-TCF, more errors can be covered for both phases of data. Based on our investigation, it can be concluded that proposed model is significantly better than other existing models and may allow the software developers to fulfill the requirements of the users operating in different environment. Future work may be extended by considering testing effort functions, different fault detection rates, and change-point phenomenon with random field environment.

REFERENCES

Chang, H., Pham, H., Lee, S.W., & Song, K.Y. (2014).A testing-coverage software reliability model with the uncertainty of operating environments.*International Journal of Systems Science: Operations and Logistics*, 1 (4), 220-227.

Chatterjee, S. and Shukla, A. (2016).Effect of test coverage and Change-point on software reliability growth based on time variable fault detection probability.*Journal of Software*, 11, 110-117.

Goel, A. and Okumoto, K. (1979). Time-dependent error-detection rate model for software reliability and other performances measures. *IEEE Transactions on Reliability*, 28 (3), 206-211.

Gokhale, S.S., Philip, T., Marinos, P.N. and Trivedi, K.S. (1996). Unification of finite failure non-homogeneous Poisson process models through test coverage. *Software Reliability Engineering, seventh International Symposium by IEEE*, 299-307.

Li, Q. and Pham, H. (2017).NHPP software reliability model considering the certainty of operating environments with imperfect debugging and testing coverage.*Applied Mathematical Modelling*, 51, 68-85.

Li, Q. and Pham, H. (2019): A generalized software reliability growth model with consideration of the uncertainty of operating environments, IEEE Access, Vol. 7, pp. 84253-84267.

Obha, M. and Yamada, S. (1984).S-Shaped software reliability growth models, In: International Colloquium on Reliability and Maintainability (Eds.), 4th, Tregastel, France, pp. 430-436.

Ohba, M. (1984).Inflection S-shaped software reliability growth model, In: Osaki S., Hatoyama Y. (Eds.) Stochastic models in reliability theory. Lecture Notes in Economics and Mathematical Systems, 235, Springer Berlin Heidelberg, pp. 144-162.

Pham, H. and Zhang, X. (2003).NHPP software reliability and cost models with testing coverage.*European Journal of Operational Research*, 145, 443-454.

Pham, H. (2014). A new software reliability model with Vtub-shaped fault detection rate and the uncertainty of operating environments.*Optimization*, 63 (10), 1481-1490.

Pham, H. (2014). Loglog fault-detection rate and testing coverage software reliability models subject to random environments.*Vietnam Journal of Computer Science*, 1 (1), 39-45.

Pham, H. (2016). A generalized fault–detection software reliability model subject to random operating environments.*Vietnam Journal of Computer Science*, 3, 145-150.

Rani, S., Agarwal, P., Jain, M. and Solanki, R. (2021): A software reliability growth model considering testing coverage subject to field environment, International Journal of Mathematics in Operational Research, Vol. 18, No. 2, pp. 145-153.

Song, K.W., Chang, L.H. and Pham, H. (2019): A testing coverage model based on NHPP software reliability considering the software operating environment and the sensitivity analysis, Mathematics 2019, Vol. 7, pp. 1-21.

Teng, X., and Pham, H. (2006).A new methodology for predicting software reliability in the random field environments, *IEEE Transactions on Reliability*, 55 (3), 458-468.

Xie, M. and Yang, B. (2003). A study of the effect of imperfect debugging on software development cost model. *IEEE Transactions on Software Engineering*, 29 (5), 471-473.

Zhang, X. and Pham, H. (2006).Software field failure rate prediction before software deployment, *Journal of Systems and Software*, 79 (3), 291-300.

Zhu, M. and Pham, H. (2020): A generalized multiple environmental factors software reliability model with stochastic fault detection process, Annals of Operations Research, 22 pp.

Zhu, M. and Pham H. (2016).A software reliability model with time-dependent fault detection and fault removal.*Vietnam Journal of Computer Science*, 3, 71-79.

Zhu, M., Zhang, X., and Pham, H. (2015).A comparison analysis of environmental factors affecting software reliability.*Journal of Systems and Software*, 109, 150-160.

Li, Q., and Pham, H. (2021): Modeling software fault-detection and fault-correction processes by considering the dependencies between fault amounts, *Applied Sciences*, 11 (15), 6998.

Yadav, N., and Yadav, V. (2023): Software reliability prediction and optimization using machine learning algorithms: A review, *Journal of Integrated Science and Technology*, 11 (1), 457-457.

Lin, J.-S., and Huang, C.-Y. (2022): Queueing-based simulation for software reliability analysis, *IEEE Access*, 10, 107729-107747.

Appendix

Week Index	**Exposure time (cumulative system test hours)**	**Fault**	**Cumulative fault**	**Week Index**	**Exposure time (cumulative system test hours)**	**Fault**	**Cumulative fault**
1	356	1	1	12	4272	2	15
2	712	0	1	13	4628	4	19
3	1068	1	2	14	4984	0	19
4	1424	1	3	15	5340	3	22
5	1780	2	5	16	5696	0	22
6	2136	0	5	17	6052	1	23
7	2492	0	5	18	6408	1	24
8	2848	3	8	19	6764	0	24
9	3204	1	9	20	7120	0	24
10	3560	2	11	21	7476	2	26
11	3916	2	13				

Table A-1: Phase I system test data

Week Index	**Exposure time (cumulative system test hours)**	**Fault**	**Cumulative fault**	**Week Index**	**Exposure time (cumulative system test hours)**	**Fault**	**Cumulative fault**
1	416	3	3	12	4992	2	25
2	832	1	4	13	5408	5	30
3	1248	0	4	14	5824	2	32
4	1664	3	7	15	6240	4	36
5	2080	2	9	16	6656	1	37
6	2496	0	9	17	7072	2	39
7	2912	1	10	18	7488	0	39
8	3328	3	13	19	7904	0	39
9	3744	4	17	20	8320	3	42
10	4160	2	19	21	8736	1	43
11	4576	4	23				

Table A-2: Phase II system test data