

Modeling software fault dynamics with imperfect debugging and time-varying fault detection

Shiv Kumar Sharma *

Rohit Kumar Rana †

Department of Mathematics

University Institute of Sciences (UIS)

Chandigarh University

Mohali 140413

Punjab

India

Abstract

In order to address time-dependent faults in software systems, this work presents a novel Imperfect Debugging Software Reliability Growth Model for time dependent faults. Recognizing the intrinsic flaws in debugging procedures, the suggested model provides a sophisticated method of evaluating software reliability. This show that the model captures the intricate dynamics of time-dependent errors through rigorous empirical validation and analytical examination. This innovative SRGM represents a significant advancement in software reliability growth modelling with imperfect debugging techniques and has great potential to improve the reliability and efficiency of real-world software systems.

Subject Classification: 90B25, 62N05.

Keywords: *Software reliability growth model, Imperfect debugging, Time-varying fault detection, Mean value function.*

Introduction

Achieving optimal reliability in software engineering is a continual issue. The debugging procedure is essential to this endeavor since it helps to fix software system bugs. Nevertheless, even if debugging plays a vital role, it is not without flaws due to human limitations, changing system needs, and incomplete comprehension of system behavior. These flaws

* E-mail: shivksharma2005@gmail.com (Corresponding Author)

† E-mail: rohitrana89996@gmail.com

present important complexity that traditional growth models of software reliability frequently fall short of capturing. Although current models provide perceptive information about the dynamics of problem identification and correction, they frequently oversimplify the complex realities of debugging techniques. As such, there is a strong need for a more advance method that recognizes and takes into account the inherent defects in the debugging procedure. The purpose of this introduction is to lay the foundation for an examination of the phenomenon of imperfect debugging processes on software reliability growth models. This will help us better understand how imperfections affect reliability assessment processes and clarify the path to more successful mitigation techniques. Ohba and Chou [1] challenged the assumption of perfect debugging, suggesting the exponential-type models could still be used with adjusted parameters. Yamada, Tokuno, and Osaki [2] included the imperfect debugging in software reliability model, providing quantitative measures for its assessment. Pradhan, Kumar, and Dhar [3] further explored the generalized inflection with S-shaped model using testing-effort function. Kapur and Younes [4] highlights the limitations of SRGM's and introduces a new model that can describe imperfect debugging processes with larger use case in various software development projects. Dong and Qin-rong [5] presents a method for improving hypothesis of prefect debugging and added imperfect debugging into it. Shyur [6] proposed a model incorporating imperfect debugging and changepoint problem to assess the dependability of the published software and identify any remaining bugs. This model's unknown parameters are estimated using the maximum likelihood method (MLE). The analysis recommends that while developing a software dependability estimation model, a changepoint in the testing process should be taken into account. Gokhale, Lyu, and Trivedi [7] modified the SRGM's premise of instantaneous and imperfect debugging, presents methods for calculating software failure rates and reliability, and includes an economic cost model to determine the best time to release the software. Finally, he introduces a rate-based simulation framework to include external debugging activities. Hanifatu, Nasiru, and Jakperik [8] showed the generalized power Weibull software reliability growth model. When compared to the current model, Raju [9] proposed model offers a somewhat better prediction up till, Inflection S-shaped Non-Homogenous Poisson Process (NHPP) growth models can benefit from the incorporation of the log-logistic TEF, and the suggested model is examined in an incomplete debugging environment. According to Ravi, Supriya, and Mohan [10], statistical process control (SPC) may effectively track software

reliability procedures and help find errors to improve software reliability. The model's applicability in the field of software reliability engineering is examined and proven. In contrast to standard SRGMs, Roy, Mahapatra, and Dey [11] suggested a modified imperfect debugging and fault generation phenomena of NHPP based SRGM that fits real failure data sets better and more precisely forecasts the behaviors of software development in the future. Iqbal [12] examined how well linear and exponential fault functions performed. The exponential fault function beats the linear one in terms of better fitting the real failure data and prediction power compared to other SRGM models. Further study is continue evolved by Li and Hoang [13] including testing coverage, multi-upgradation SRGM given by Chatterjee, Saha, and Sharma [14]. The recent studies [15, 16, 17] collectively highlight the need to account for imperfect debugging in SRGM. Kaur and Sharma [18] described about the SRGM in terms of Neural Networks. The proposed model aims to address the limitations of traditional perfect debugging models by incorporating the complexities of imperfect debugging processes. This paper proposes a Mean Value Function (MVF) to characterize the software reliability growth in the presence of imperfect debugging.

Proposed Model and Analysis

Many techniques exist in literature to predict the faults, our proposed model here considering [17] taking evaluation matrix as sum of squared error (SSE), root mean squared error (RMSE) and R-squared. Our proposed model mean value function (MVF) is defined as:

$$MVF = \frac{a \left(1 - e^{-a \left(1 - e^{-\left(\frac{-a r^2}{2} \right)} \right)} \right)}{b * t}$$

Here, $a(t)$: Initial fault detection rate. This indicates how quickly faults are initially identified in the software. Higher rates suggest proactive fault detection, improving software reliability, $b(t)$: Fault detection rate under perfect debugging. This shows how efficiently faults are detected and fixed under ideal conditions. Higher rates mean faster resolution of issues and better software reliability, $\alpha(t)$: Fault introduction rate. This represents the speed at which new faults arise over time. Higher rates increase the likelihood of new defects, potentially lowering software reliability, r : probability that fault will be imperfectly debugged. This reflects the chance that detected faults aren't completely resolved during debugging.

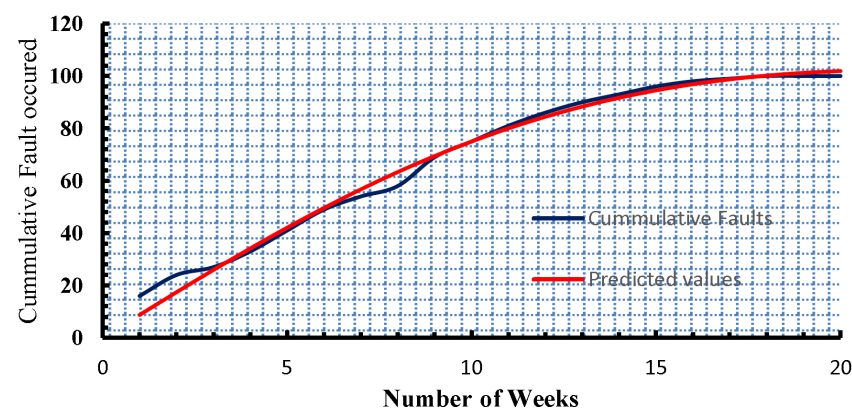


Figure 1

Comparison of Predicted values vs Actual cumulative faults in software

Higher probabilities mean some faults may persist. β : is fault removal rate. This denotes how quickly faults are eliminated through debugging. Higher rates lead to faster defect resolution and improved software reliability. Considering dataset as mentioned in [17] and estimating the parameter values of MVF we get, $a = 99.99999454$, $b = 0.01857718$, $r = 0.99155500$, $\alpha = 0.98845721$ and $\beta = 0.00332224$ for dataset-1 the values obtained is shown in Figure 1. Now we will take values of parameter for comparison of the evaluation parameters to make the ranking of proposed model based on other model performance as follows:

Table 1

Comparison and Ranking of Mean values of parameter's

Model Name	SSE	RMSE	R-squared	Rank Index	Rank
Proposed Model	152.8518	1.1528	0.99216	0.9981	1
P-N-Z Model	179.5843	3.2502	0.9877	0.99	2
Zhang-Teng-Pham	176.4086	3.5497	0.9853	0.967	3
P-Z Model	229.6817	3.9131	0.9821	0.8603	4
Yamada Imperfect debugging M2	232.3422	3.5928	0.9849	0.8829	5
Yamada exponential	313.1996	4.4244	0.9771	0.7583	6
Yamada Rayleigh	790.7322	7.03	0.9422	0.5426	7

As seen in Table 1, Proposed model has performed effectively well then other models. The actual fault is 110 and 102 is predicted fault thus we get remaining faults for imperfect debugging is 8.

Conclusion

This paper has presented the efficacy of the suggested model of Imperfect Debugging Software Reliability Growth Models in accurately predicting the expected cumulative number of faults within software systems. Through comprehensive analysis, it has been observed that the values reported in Table 1 across datasets closely approximate the actual fault counts. Thus, it is evident that while working with Imperfect Debugging SRGMs introduces added complexity, the predictive values yielded are significantly more accurate compared to traditional perfect debugging models. Moreover, our proposed model has surpassed the performance of existing Imperfect Debugging models documented yet. This substantiates the potential of Imperfect Debugging SRGMs as a viable approach for enhancing reliability assessment in software engineering practices.

References

- [1] M. Ohba and X. M. Chou, "Does imperfect debugging affect software reliability growth?" in *Proc. 11th Int. Conf. Software Engineering (ICSE)*, pp. 237–244 (1989), doi: 10.1145/74587.74619.
- [2] S. Yamada, K. Tokuno, and S. Osaki, "Software reliability measurement in imperfect debugging environment and its application," *Reliability Engineering & System Safety*, vol. 40, no. 2, pp. 139–147 (1993).
- [3] V. Pradhan, A. Kumar, and J. Dhar, "Enhanced growth model of software reliability with generalized inflection S-shaped testing-effort function," *Journal of Interdisciplinary Mathematics*, vol. 25, no. 1, pp. 137–153 (2022).
- [4] P. K. Kapur and S. Younes, "Modelling an imperfect debugging phenomenon in software reliability," *Microelectronics Reliability*, vol. 36, no. 5, pp. 645–650 (1996), doi: 10.1016/0026-2714(95)00157-3.
- [5] Y. Dong and Y. Qin-rong, "A software reliability growth model under imperfect debugging," *Journal-Wuhan University Natural Sciences Edition*, vol. 45, no. 5B, pp. 703–705 (1999).

- [6] H.-J. Shyur, "A stochastic software reliability model with imperfect-debugging and change-point," *Journal of Systems and Software*, vol. 66, no. 2, pp. 135–141 (2003), doi: 10.1016/S0164-1212(02)00071-7.
- [7] S. S. Gokhale, M. R. Lyu, and K. S. Trivedi, "Incorporating fault debugging activities into software reliability models: A simulation approach," *IEEE Trans. Rel.*, vol. 55, no. 2, pp. 281–292 (2006), doi: 10.1109/TR.2006.874911.
- [8] M. N. Hanifatu, S. Nasiru, and D. Jakperik, "Generalized power Weibull software reliability growth model," *Journal of Information and Optimization Sciences*, vol. 43, no. 8, pp. 1987–2006 (2022).
- [9] O. N. Raju, "Software reliability growth models for the safety critical software with imperfect debugging," *Int. J. Comput. Sci. Eng.*, vol. 3, no. 8, pp. 3019–3026 (2011).
- [10] S. P. Ravi, N. Supriya, and G. K. Mohan, "SPC for software reliability: Imperfect software debugging model," *Int. J. Comput. Sci. Issues*, vol. 8, no. 3, pp. 219 (2011).
- [11] P. Roy, G. S. Mahapatra, and K. N. Dey, "An NHPP software reliability growth model with imperfect debugging and error generation," *Int. J. Rel. Qual. Saf. Eng.*, vol. 21, no. 2, Art. no. 1450008 (2014).
- [12] J. Iqbal, "Software reliability growth models: A comparison of linear and exponential fault content functions for study of imperfect debugging situations," *Cogent Engineering*, vol. 4, no. 1, Art. no. 1286739 (2017).
- [13] Q. Li and P. Hoang, "NHPP software reliability model considering the uncertainty of operating environments with imperfect debugging and testing coverage," *Applied Mathematical Modelling*, vol. 51, pp. 68–85 (2017).
- [14] S. Chatterjee, D. Saha, and A. Sharma, "Multi-upgradation software reliability growth model with dependency of faults under change point and imperfect debugging," *Journal of Software: Evolution and Process*, vol. 33, no. 6, Art. no. e2344 (2021), doi: 10.1002/smr.2344.
- [15] S. K. Sharma, K. Thukral, and R. K. Rana, "A supervised learning approach with ANN for software reliability growth modeling," *Quality and Reliability Engineering International*, pp. 1-7 (2026), doi: 10.1002/qre.70229.

- [16] Z. Liu and K. Rui, "Imperfect debugging software belief reliability growth model based on uncertain differential equation," *IEEE Trans. Rel.*, vol. 71, no. 2, pp. 735–746 (2022).
- [17] P. Panwar and R. Kaur, "Effect of imperfect debugging on prediction of remaining faults in software," in *Proc. 5th Int. Conf. Soft Computing for Problem Solving (SocProS 2015)*, vol. 1, Singapore: Springer (2016).
- [18] H. Kaur and S. K. Sharma, "Exploration on reliability theory using LGCM model in neural network," *Adv. Math. Sci. J.*, vol. 8, pp. 5349–5359 (2020).

Received February, 2026