Software Fault Prediction Using Machine Learning

Tushar Arora
Maharaja Agrasen
Institute of Technology
tushararora1401@gmail.com

Harshit Saini
Maharaja Agrasen
Institute of Technology
harshitsaini3582@gmail.com

ABSTRACT

The IT and software industry has grown tremendously over the past few years, creating an increasing impact on the lives of people and on society as a whole. Consequently, we must make the software and applications more accurate, free of major errors, and more reliable. Therefore, predicting software flaws could be very useful in the IT field and will have a profound impact on society at large.

1. INTRODUCTION

The vast area of software development and different applications makes it challenging for software developers and also customers to observe, maintains and manage software applications. Moreover, the fourth industrial revolution employs artificial intelligence by software industry is one of the promising sectors of modern times that observes a constant transformation in its practices because of the automating large quantities of software technologies [1]. The size and complexity of current software is increasing day by day. As a result, software engineers are struggling continuously with faults from the beginning of the development phase.

The classification of the software faults is important in real-time, otherwise, the effort and cost of finding defects hiding in an application are also rising fast. This inspires the development of automated fault prediction models for software fault prediction that can forecast the software defects. If software defects are identified before the release of software that can help the developer to allocate and fix those defect modules easily.

2. MATERIALS & METHODS

2.1 Data Collection

In this project, we have used 3 open source publicly available data from PROMISE Software Engineering Database. These datasets Tim Menzies et al. have been used in their research paper [1]. In another study, Jureczko et al. [2] have been assembled a software fault prediction model to predict the software defects using machine learning algorithms. They have discussed in their paper about 8 projects (PROMISE Repository) data and by taking 19 CK metrics and McCabe metrics for constructed a predictive model. In our study, we have used 22 attributes for building our automated fault predict model. Table 1 shows 22 different attributes from software defect datasets including 21 independent metrics and one is outcome information. i.e. which is faulty and no-fault. We are using JM1, CM1, PC1 datasets which were implemented in C language.

2.2 Classification Techniques

Machine learning algorithm has been creating a significant role in software engineering fields. In recent years, machine learning techniques are one of the most operational techniques what are gained significantly high performance in real-world problems for the research and technical community. Harshita et al. [31] discussed in their review, there are common use of machine learning techniques for constructing software fault prediction models such as fuzzy logic-based software defect prediction, Naïve Bayes (NB), neural network (NN), random forest (RF), support vector machine (SVM), P-SVM, k-nearest neighbors (KNN), etc. Ruchita Malhotra [3] described in her systematic mapping study, the top five machine learning techniques were used to software defect analysis such as DT (46%), NB (74%), MLP in NN (85%), RF (59%), SVM (27.7%), etc.

In this study, 6 machine learning (ML) techniques have been considered to construct the defect model: Decision Tree (DT), k-nearest neighbors (KNN), Logistics Regression (LR), Naïve Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM).

2.3 Algorithms

3. RESULTS & DISCUSSION

4. CONCLUSION

5. ACKNOWLEDGEMENTS

6. REFERENCES

[1] H. B. Bolat, G. T. Temur, and IGI Global, Agile approaches for successfully managing and executing projects in the fourth industrial revolution.

[2] G. P. Bhandari and R. Gupta, “Machine learning based software fault prediction utilizing source code metrics,” in 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS),

2018, pp. 40–45.

[3] R. Malhotra, “A systematic review of machine learning techniques for software fault prediction,” Appl. Soft Comput., vol. 27, pp. 504–518, Feb. 2015.

[4] L. Son et al., “Empirical Study of Software Defect Prediction: A Systematic Mapping,” Symmetry (Basel)., vol. 11, no. 2, p. 212, Feb. 2019.

[5] C. W. Yohannese and T. Li, “A Combined-Learning Based Framework for Improved Software Fault Prediction,” Int. J. Comput. Intell. Syst., vol. 10, no. 1, p. 647, Dec. 2017.

[6] A. Hudaib et al., “ADTEM-Architecture Design Testability Evaluation Model to Assess Software Architecture Based on Testability Metrics,” J. Softw. Eng. Appl., vol. 08, no. 04, pp. 201–210, Apr. 2015.

[7] S. Elmidaoui, L. Cheikhi, and A. Idri, “Towards a Taxonomy of Software Maintainability Predictors,” Springer, Cham, 2019, pp. 823–832.

[8] M. Riaz, E. Mendes, and E. Tempero, “A systematic review of software maintainability prediction and metrics,” in 2009 3rd International Symposium on Empirical Software Engineering and Measurement, 2009, pp. 367–377.

[9] “PROMISE DATASETS PAGE.” [Online]. Available: http://promise.site.uottawa.ca/SERepository/datasets-page.html. [Accessed: 01-Jul-2019].

[10] R. Malhotra and S. Kamal, “An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data,” Neurocomputing, vol. 343, pp. 120–140, May 2019.

[11] 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology. .

[12] P. Oman and J. Hagemeister, “Construction and testing of polynomials predicting software maintainability,” J. Syst. Softw., vol. 24, no. 3, pp. 251–266, Mar. 1994.

[13] T. Anderson, P. A. Barrett, D. N. Halliwell, and M. R. Moulding, “Software Fault Tolerance: An Evaluation,” IEEE Trans. Softw. Eng., vol. SE-11, no. 12, pp. 1502–1510, Dec. 1985.

[14] I. U. Nisa and S. N. Ahsan, “Fault prediction model for software using soft computing techniques,” in 2015 International Conference on Open Source Systems & Technologies (ICOSST), 2015, pp. 78–83.

[15] S. N. Ahsan and F. Wotawa, “Fault Prediction Capability of Program File’s Logical-Coupling Metrics,” in 2011 Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement, 2011, pp. 257–262.

[16] D. Radjenović, M. Heričko, R. Torkar, and A. Živkovič, “Software fault prediction metrics: A systematic literature review,” Inf. Softw. Technol., vol. 55, no. 8, pp. 1397–1418, Aug. 2013.

[17] R. Malhotra, “Comparative analysis of statistical and machine learning methods for predicting faulty modules,” Appl. Soft Comput., vol. 21, pp. 286–297, Aug. 2014.

[18] Z. Xu et al., “Software defect prediction based on kernel PCA and weighted extreme learning machine,” Inf. Softw. Technol., vol. 106, pp. 182–200, Feb. 2019.

[19] J. Moeyersoms, E. Junqué de Fortuny, K. Dejaeger, B. Baesens, and D. Martens, “Comprehensible software fault and effort prediction: A data mining approach,” J. Syst. Softw., vol. 100, pp. 80–90, Feb. 2015.

[20] Q. Yu, J. Qian, S. Jiang, Z. Wu, and G. Zhang, “An Empirical Study on the Effectiveness of Feature Selection for Cross-Project Defect Prediction,” IEEE Access, vol. 7, pp. 35710–35718, 2019.

[21] J. Hotzkow and Jenny, “Automatically inferring and enforcing user expectations,” in Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis - ISSTA 2017, 2017, pp. 420–423.

[22] A. A. Hudaib, H. N. Fakhouri, A. A. Hudaib, and H. N. Fakhouri, “An Automated Approach for Software Fault Detection and Recovery,” Commun. Netw., vol. 08, no. 03, pp. 158–169, Jul. 2016.

[23] S. Montani and C. Anglano, “Achieving self-healing in service delivery software systems by means of case-based reasoning,” Appl. Intell., vol. 28, no. 2, pp. 139–152, Apr. 2008.

[24] “A New Way to Find Bugs in Self-Driving AI Could Save Lives - IEEE Spectrum.” [Online]. Available: https://spectrum.ieee.org/tech-talk/robotics/artificial-intelligence/better-bug-hunts-in-selfdriving-car-ai-could-save-lives. [Accessed: 02-Jul-2019].

[25] C. Catal and B. Diri, “A systematic review of software fault prediction studies,” Expert Syst. Appl., vol. 36, no. 4, pp. 7346–7354, May 2009.

[26] V. U. B. CHALLAGULLA, F. B. BASTANI, I.-L. YEN, and R. A. PAUL, “EMPIRICAL ASSESSMENT OF MACHINE LEARNING BASED SOFTWARE DEFECT PREDICTION TECHNIQUES,” Int. J. Artif. Intell. Tools, vol. 17, no. 02, pp. 389–400, Apr. 2008.

[27] E. Shihab et al., “Studying re-opened bugs in open source software,” Empir. Softw. Eng., vol. 18, no. 5, pp. 1005–1042, Oct. 2013.

[28] M. Shepperd, Q. Song, Z. Sun, and C. Mair, “Data Quality: Some Comments on the NASA Software Defect Datasets,” IEEE Trans. Softw. Eng., vol. 39, no. 9, pp. 1208–1215, Sep. 2013.

[29] T. Menzies, J. Distefano, A. O. S, and R. M.

Chapman, “Assessing Predictors of Software Defects.”

[30] M. Jureczko and L. Madeyski, “Towards identifying software project clusters with regard to defect prediction,” in Proceedings of the 6th International Conference on Predictive Models in Software Engineering - PROMISE ’10, 2010, p. 1.

[31] H. Tanwar and M. Kakkar, “A Review of Software Defect Prediction Models,” Springer, Singapore, 2019, pp. 89–97.

[32] S. Kanmani, V. R. Uthariaraj, V. Sankaranarayanan, and P. Thambidurai, “Object-oriented software fault prediction using neural networks,” Inf. Softw. Technol., vol. 49, no. 5, pp. 483–492, May 2007.

[33] P. N. Tan, M. Steinbach, and V. Kumar, Introduction to data mining. Pearson Addison Wesley, 2005.

[34] S. García, J. Luengo, and F. Herrera, Data Preprocessing in Data Mining, vol. 72. Cham: Springer International Publishing, 2015.