International Journal of Information Processing, 8(2), 1-12, 2014
ISSN : 0973-8215
IK International Publishing House Pvt. Ltd., New Delhi, India

# Framework to Assess the Effectiveness of Software Fault Classification

Yeresime Suresh[a], Lov Kumar[a], Santanu Kumar Rath[a]

[a]Department of Computer Science and Engineering, National Institute of Technology, Rourkela - 769008, Contact: suresh.vec04@gmail.com, lovkumar505@gmail.com, skrath@nitrkl.ac.in

Software fault prediction is helpful in deciding the amount of effort needed for software development. In literature it is observed that, many models have been studied and evaluated on software products to determine best suitable model for fault prediction based on certain performance criteria. However very less significant work has been done on feasibility of fault prediction model. In this article, a cost evaluation framework has been proposed which performs cost based analysis for mis-classification of faults. Appropriately, this work focuses on inspecting the usability of fault prediction. This paper highlights on classification of faults on the basis of Radial Basis Function Network (RBFN) and it's application on Apache Integration Framework (AIF) version 1.6. Fault prediction is found to be useful where normalized estimated fault removal cost (NEcost) was less than certain threshold value. Gradient RBFN model obtained promising results in terms of cost analysis when compared with those of basic RBFN and hybrid RBFN models.

**Keywords :** Apache Integration Framework, CK Metric Suite, Cost, Fault, Neural Network, Radial Basis Function.

## 1. INTRODUCTION

Fault prediction is necessary in software development life cycle and is carried out to identify fault-prone modules. Fault prediction not only gives an insight to the need for increased quality of monitoring during software development but also provides necessary tips to undertake suitable verification and validation approaches that eventually lead to improvement of in terms of efficiency and effectiveness of fault prediction. Effectiveness of a fault-prediction is studied by applying a part of previously known data related to faults and predicting its performance against other part of the fault data.

Several researchers have worked on building prediction models for software fault prediction but less emphasis has been given on the study of effectiveness of fault prediction. This paper aims to assess the influence of RBFN model in predicting faults by using CK metrics as requisite input to the prediction models to put the results of a fault-prediction technique in proper perspective. Also an attempt has been made to assess the influence of fault removal cost in predicting whether performing fault prediction analysis is useful or not. Radial Basis Function Network (RBFN) is applied as a classifier to classify faulty, not-faulty classes and predict the fault detection accuracy rate respectively.

The rest of the paper is organized as follows: Section 2, summarizes on software metrics and their usage in fault classification. Section 3, highlights on research background. Section 4 describes the proposed work for fault classification by applying RBFN and also presents a framework for emphasizing the effectiveness of fault classification based on cost parameter. Section 5 highlights on the parameters used for evaluating the performance of each of the applied RBFN model. Section 6 presents the results and analysis about fault classification. Section 7 concludes the paper with scope for future work.

Table 12
Fault Removal Cost for AIF 1.6

| | Precision | TP Rate | FP Rate | TN Rate | FN Rate | Accuracy (%) | Ecost | Tcost | NEcost |
|---|---|---|---|---|---|---|---|---|---|
| Basic RBFN | 25.71 | 47.87 | 33.46 | 66.54 | 52.13 | 62.90 | 3622.3 | 3546.8 | 1.0213 |
| Gradient RBFN | 99.19 | 64.89 | 0.13 | 99.87 | 35.11 | 93.05 | 2922 | 3546.8 | 0.8238 |
| Hybrid RBFN | 40 | 10.64 | 3.86 | 96.14 | 89.36 | 79.40 | 3162.8 | 3546.8 | 0.8917 |

fault classification analysis. Here normalized data set of CK metrics suite was used as requisite input to the prediction models.

Table 12 shows that gradient model of RBFN obtained better fault classification when compared with the other two RBFN models. In terms of cost evaluation, it can be concluded that Gradient RBFN has low value of NEcost when compared with that of Basic RBFN and Hybrid RBFN. So from the proposed work of cost evaluation framework, it can be noted that it is better to perform fault prediction when NEcost is less than one.

In future, work should be replicated to other open source projects like Mozilla using different AI techniques to analyze the model used in achieving higher accuracy of fault prediction is effective or not.

## REFERENCES


1. McCabe T J. A Complexity Measure, *IEEE Transactions on Software Engineering*, 2(4):308-320, December 1976.
2. Halstead M H. Elements of Software Science. *New York, NY, USA: Elsevier Science*, 1977.
3. Li W and Henry S. Maintenance Metrics for the Object-Oriented Paradigm, in *Proceedings of* $1^{st}$ *International Software Metrics Symposium*, pages 52–60, 1993.
4. Chidamber S R and Kemerer C F. A Metrics Suite for Object-Oriented Design, *IEEE Transactions on Software Engineering*, 20(6):476-493, June 1994.
5. Abreu F B E and Carapuca R. Object-Oriented Software Engineering: Measuring and Controlling the Development Process, in *Proceedings of the* $4^{th}$ *International Conference on Software Quality*, pages 1-8, 1994.
6. Lorenz M and Kidd J. *Object-Oriented Software Metrics*. NJ, Englewood: Prentice Hall, 1994.
7. Martin R. OO Design Quality Metrics - An Analysis of Dependencies, in *Proceedings Workshop Pragmatic and Theoretical Directions in Object-Oriented Software Metrics (OOPSLA94)*, 1994.
8. Tegarden D P, Sheetz S D and Monarchi D E. A Software Complexity Model of Object-Oriented Systems, *Decision Support Systems*, 13(4):241–262, 1995.
9. Melo W L and Abreu F B E. Evaluating the Impact of Object-Oriented Design on Software Quality, in *Proceedings of the* $3^{rd}$ *International Software Metrics Symposium, Berlin*, pages 90–99, 1996.
10. Briand L, Devanbu P, and Melo W L. An Investigation into Coupling Measures for C++, in *Proceedings of International Conference on Software Engineering Association for Computing Machinery*, pages 412–421, 1997.
11. Etzkorn L, Bansiya J and Davis C. Design and Code Complexity Metrics for OO Classes, *Object-Oriented Programming*, pages 35–40, April 1999.
12. Basili V R, Briand L C and Melo W L. A Validation of Object-Oriented Design Metrics as Quality Indicators, *IEEE Transactions on Software Engineering*, 22(10):751–761, October 1996.
13. Briand L C, Wust J, Daly J W and Poter D V. Exploring the Relationships between Design Measures and Software Quality in Object-Oriented Systems, *Journal of Systems and Software*, 51(3):245–273, May 2000.
14. Tang M H, Kao M H and Chen M H. An Empirical Study on Object-Oriented Metrics, *in Proceedings of* $6^{th}$ *International Conference on Software Metrics Symposium*, pages 242–249, 1999.

15. Emam K E, Melo M and Machado J C. The Prediction of Faulty Classes using Object-Oriented Design Metrics," *Journal of System and Software*, 56(1):63–75, 2001.
16. Khoshgoftaaar T M, Allen E B, Hudepohl J P and Aud S J. Application of Neural Networks to Software Quality Modeling of a Very Large Telecommunications Systems, *IEEE Transactions on Neural Networks*, 8(4):902–909, 1997.
17. Hochman R, Khoshgoftar T M, Allen E B and Hudepohl J P. Evolutionary Neural Networks: A Robust Approach to Software Reliability Problems, in *Proceedings of the Eight International Symposium on Software Reliability Engineering*, pages 13–26, USA, November 1997.
18. Menzies T, Caglayan B, Kocaguneli E, Krall J, Peters F and Turhan B. The PROMISE Repository of Empirical Software Engineering Data, West Virginia University, Department of Computer Science, 2012. ({http://promisedata.googlecode.com})
19. Yogendra Kumar Jain and Santosh Kumar Bhandare. Min-Max Normalization Based Data Perturbation Method for Privacy Protection, *International Journal of Computer and Communication Technology*, 2(8):45–50, 2011.
20. Kanmani S and Uthariaraj V Rymend. Object-Oriented Software Quality Prediction using General Regression Neural Networks, *SIGSOFT Software Engineering Notes*, 29(5):1–6, New York , USA, September 2004.
21. Nachiappan Nagappan and Williams Laurie. Early Estimation of Software Quality using In-process Testing Metrics: A Controlled Case Study, in *Proceedings of the Third Workshop on Software quality*, pages 1–7, St. Louis, Missouri, USA, 2005.
22. Hector M Olague, Letha H Etzkorn, Sampson Gholston and Stephen Quattlebaum. Empirical Validation of Three Software Metrics Suites to Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Processes, *IEEE Transactions on Software Engineering*, 33(6):402-409 , Piscataway, NJ, USA, June 2007.
23. Aggarwal K K, Yogesh Singh, Arvinder Kaur and Ruchika Malhotra. Empirical Analysis for Investigating the Effect of Object-Oriented Metrics on Fault Proneness: A Replicated Case Study, *Software Process Improvement*, 14(1):39–62, 2009.
24. Fangjun Wu. Empirical Validation of Object-Oriented Metrics on NASA for Fault Prediction, in *Proceedings of International Conference on Advances in Information Technology and Education*, pages 168–175, 2011.
25. Heena Kapila and Satwinder Singh. Article: Analysis of CK Metrics to Predict Software Fault-Proneness using Bayesian Inference, *International Journal of Computer Applications*, 74(2):1–4, July 2013.
26. Jiang Y, Cukic B and Ma Y. Techniques for Evaluating Fault Prediction Models, *Empirical Software Engineering*, 13(5):561–595, 2008.
27. Mende T and Koschke R. Revisiting the Evaluation of Defect Prediction Models, in *Proceedings of the 5th International Conference on Predictor Models in Software Engineering (PROMISE)*, vol 7, pages 1–10, New York, USA, 2009.
28. Mende T and Koschke R. Effort-aware Defect Prediction Models, in *Proceedings of 14th IEEE European Conference on Software Maintenance and Re-engineering (CSMR)*, pages 107–116, March 2010.
29. Arisholm E, Briand L C and Johannessen E B. A Systematic and Comprehensive Investigation of Methods to Build and Evaluate Fault Prediction Models, *Journal of System and Software*, 83(1):2–17, January 2010.
30. Wagner S. A Literature Survey of the Quality Economics of Defect-detection Techniques", in *Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering (ISESE)*, pages 194–203, New York, USA, 2006.
31. Jones C. Software Quality in 2010: A Survey of the State of the Art, Founder and Chief Scientist Emeritus, December 2010.
32. Ross Huitt and Norman Wilde. Maintenance Support for Object-oriented Programs, *IEEE Transactions on Software Engineering*, 18(12):1038–1044, December 1992.
33. C Cagatay, Performance Evaluation Metrics for Software Fault Prediction Studies, *Acta Polytechnica Hungarica*, 9(4):193–206, 2012.
34. X Yuan, Khoshgoftaar, E B Allen and K Ganesan. Application of Fuzzy Clustering to Software Quality Prediction, in *Proceedings of 3rd Symposium on ASSEST*, pages 85-91, 2000.

**Yeresime Suresh** was born in Bellary, India, on February 20, 1985. He received his MTech Degree in Software Engineering from Sri Jayachamarajendra College of Engineering, Mysore, India in 2010. He is presently working towards the Ph.D Degree in Computer Science at National Institute of Technology, Rourkela, India. His areas of interest include Software Metrics, Software Fault Prediction, Object-Oriented Testing.

**Lov Kumar** did his BTech in Computer Science and Engineering from C V Raman College of Engineering, Bhubaneswar. Currently he is pursuing his MTech in Computer Science and Engineering with Software Engineering specialization at NIT, Rourkela. His areas of interest are in Software Quality Estimation using Software Metrics.

**Dr. Santanu Kumar Rath** received his Ph.D Degree from Indian Institute of Technology, Kharagpur, India. He is currently a Professor in Department of Computer Science and Engineering, NIT, Rourkela, India since 1988. He has authored and co-authored many research papers. His research interests are Software Engineering, Bioinformatics, System Engineering and Management. Dr. Santanu Kumar Rath is a Senior Member of the IEEE, USA and ACM, USA and Petri Net Society, Germany and is a professional member of MIEEE, MACM, FIE and FCSI.