IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Role of CK Metrics to Identify Fault-Proneness in Object Oriented Software: A Survey

Sunil Sikka[1], Utpal Shrivastava[2], Pooja[3*]

[1,2,3] Dept. of Computer Science and Engineering, Amity University Haryana, Gurugram, India

**Corresponding Author: pooja1894jangra@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 23/May/2018, Published: 31/May/2018

Abstract- Predicting Fault-proneness of software modules is the essential for cost effective test planning. Various studies have shown the importance of software metrics in predicting fault-proneness of the software.Chidamber and Kemerer (CK) metrics suite is the most widely used metrics suite for the purpose of object-oriented software fault-proneness prediction. The current paper is aimed to review various studies available in literature to predict software fault-proneness using CK metrics.

Keywords- Fault-proneness,CK metrics.

I. INTRODUCTION

Fault-proneness of a software module predicts the probability of the presence of faults in it. Fault-proneness could play a key role in quality control of software [1]. Finding faults during the software testing phase is costly and time consuming. The cost effective approach is to estimate and prevent faults at the early stage of software development. It also helps to systematically plan the testing process in advance. Currently majority of the software development is based on object-oriented approach. Fundamental features of object- oriented approach i.e. Encapsulation, Inheritance, coupling and cohesion are key factors to determine fault-proneness of classes [2].

Object-oriented metrics play a very important role toquantify the effect of key factors to determine the fault- proneness. Most commonly used metrics includes the CK metrics suite which includes six metrics Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT),Lack of Cohesion in Methods(LCOM), Number of Children (NOC), Response for Class (RFC), and Coupling Between Object classes(CBO) [3].WMC of a class is the weighted sum of all the methods defined in a class.DIT of a class is the maximum length from the node indicating that class to the root of the inheritance hierarchy.RFC of a class is defined as set of methods that can be potentially executed in response to a message received by an object of that class.NOC of a class is the count of the number of immediate child classes that have inherited from a given class.CBO of a class is the count of the number of other classes to which it is coupled. LCOM of a class it is the difference between number of pairs of distinct methods that do not share same instance variable and number of pairs of distinct methods that share same instance variable.

This paper is aimed to conduct a review of various studies that aimed to predict fault-proneness of object-oriented software using CK metrics.The rest of the paper is presented into three sections. Section 2 presents the various techniques used for software fault-prediction. Section 3 presents related work in field of fault-proneness prediction using CK metrics. Finally section 4 presents the conclusion of the paper.

II. TECHNIQUES FOR FAULT-PRONENESS PREDICTION

Various techniques are available that are used to predict the fault-proneness of software modules. Many researchers have used these techniques to build various fault-prediction models where software metrics are used as independent variables and fault proneness as dependent variable [4,5,6,2,7,8,9,10]. Brief description of these techniques is as follows.

a) Logistic Regression (LR): LR is the most commonly used technique in literature to predict dependent variable from set of independent variables.It is of two types Univariate LR and Multivariate LR. Univariate LR is a statistical method that formulates a mathematical model showcasing relationship among each independent and dependent variable. Multivariate LR is used to build a prediction model where multiple independent variables are used to determine the collective effect on dependent variable. The main advantage of LR is to handle nonlinear relationship between dependent and independent variables [11,12].

b) Linear Regression: This technique is used to establish only linear relationship between independent and dependent variables by fitting a best line. Linear

Regression is of mainly two types Simple Linear Regression and Multiple Linear Regression. Simple Linear Regression is characterized by one independent variable and Multiple Linear Regression is characterized by multiple (more than 1) independent variables.The main advantage of this technique is that it is very simple method. But on the other hand it only models relationships between dependent and independent variables that are linear which is not always possible [13,14].

c) Naive Bayes (NB): Naive Bayes classification is a supervised machine learning technique. It is simple but one of the most effective techniques of the classification. This technique is based on BayesTheoremandassumes that every pair of features being classified is independent of each other. The main advantage of this classifier is that it only requires a small number of training data [15,16].

d) Principal Component Analysis (PCA): PCA is used as a dimension-reduction toolwhich can reduce a large set of variables to a small set that still contains most of the information in the large set. The main advantage of PCA is lack of redundancy of data given the orthogonal components and the disadvantage is the difficulty in accurate evaluation of covariance matrix while applying PCA [17,18].

III. RELATED WORK

Many authors have investigated the effect of CK metrics on software fault-proneness. Some of the studies available in the literature are presented here.

Basili et al. [19] conducted an empirical validation of CK metrics to investigate the prediction of fault-proneness of a class. Authors concluded that CBO, NOC, RFC, DIT and WMC are significantly correlated with fault- proneness.Ping Yu et al. [20] investigated the relationship between the fault-proneness and ten metrics including CK metrics. Authors concluded that LOC, CBO, NOC and RFC metrics to be the best in predicting the fault-proneness of classes than LCOM and DIT metrics. Gyimothy et al. [18] validated CK metrics for fault-proneness detection using the source code of the open source Web and e-mail suite called Mozilla. For fault-proneness detection authors used Regression and Machine Learning methods to validate the metrics for fault- proneness prediction. It is found that CBO and LOC metrics to be the best in predicting the fault-proneness of classes but DIT metric is untrustworthy and NOC cannot be used at all for fault-proneness prediction. Yuming et al. [1] also explored the relationship between metrics and fault-proneness. Authors found that CBO, WMC, RFC and LCOM metrics are much better than DIT for any fault severity. Hector M. Olague et al. [21] empirically validated three metric suits (CK, QMOOD, MOOD) to predict fault-proneness for highly iterative and agile software development process over multiple releases of software and concluded that CK metrics have been better and more reliable predictor of fault proneness than the MOOD and QMOOD metrics. Aggarwal et al. [6] conducted an empirical validation of twenty six metrics including CK metrics to investigate the effect of individual and combined metrics on fault proneness. The study established the relationship among various metrics and found many metrics provide redundant information. Yuming Zhou et al. [1] investigated the ability of complexity metrics to predict fault-prone classes and concluded that LOC and WMC are better fault predictors than standard deviation method complexity (SDMC) and average method complexity (AMC) metric.Malhotra et al. [7] investigated the relationship between nineteen object oriented metrics including CK metrics and fault-proneness of a class. Out of nineteen metrics eight metrics LOC, WMC, CBO, RFC, Number of public Methods (NPM), Cause & Effect (CE), Average Method Complexity (AMC) and Cohesion Among Methods of Class (CAM) found the best predictors of fault proneness.

IV. CONCLUSION

This research paper reviewed many studies carried out by various researchers for fault prediction using CK metrics. The review shows that some of the CK metrics are good predictor of fault-proneness of classes. As per many reviews LCOM and DIT are good predictor. Also various types of techniques are discussed which are used for fault prediction in the literature.

References

[1] Yuming Zhou, BaowenXu, Hareton Leung, "On the Ability of Complexity Metrics to Predict Fault-Prone Classes in Object-Oriented Systems", Journal of Systems and Software, Vol. 83, Issue 4, April 2010.

[2] SuraiyaParveen& R A Khan: "Fault Proneness Model for Object-Oriented Software" Design Phase Perspective. 2009 International Conference on Computer Engineering and Application.IACSIT Press, Singapore.

[3] S. Chidamber and C.kemerer, "A metrics suite for object oriented design", IEEE Transaction on Software Engineering, Vol. 20, No. 6, June 1994.

[4] Tibor Gyimothy, Rudolf Ferenc, and IstvanSiket, "Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction", IEEE Transaction on Software Engineering, Vol 31, No. 10, October 2005.

[5] Yuming Zhou and Hareton Leung, "Empirical analysis of object oriented design metrics for predicting high severity faults", IEEE Transactions on Software Engineering, Vol. 32, Issue 10, 2006.

[6]K.K.Aggarwal, Yogesh Singh, ArvinderKaur, RuchikaMalhotra,"Investigating effect ofDesign Metrics on

Fault Proneness in Object-Oriented Systems", Journal of Object Technology, December 2007.
[7] Ruchika Malhotra, YogeshSingh,"On the Applicability of Machine Learning Techniques for Object Oriented Software Fault Prediction", Department of Software Engineering, Delhi Technological University, Delhi, India. International Journal (SEIJ), September 2011.
[8] HeenaKapila, Satwinder Singh, "Analysis of CK Metrics to Predict Software Fault-Proneness using Bayesian Inference", International Journal of Computer Applications July 2013.
[9]Alexandre Boucher, MouradBadri, "Using Software Metrics Thresholds to Predict Fault-Prone Classes in Object-Oriented Software",2016 4th Intl Conf on Applied Computing and Information Technology.
[10] Jehad Al Dallal , SandroMorasca,"Investigating the Impact of Fault Data Completeness over Time on Predicting Class Fault-Proneness", INFSOF 5907, 7 November 2017.
[11] Chao-Ying Joanne Peng, Kuk Lida Lee, Gary M. Ingersoll,"An Introduction To Logistic RegressionAnalysis And Reporting", The Journal Of Educational Research, September/October 2002 ,Vol. 96,No. 1.
[12] Murat Korkmaz, SelamiGüney, ŞuleYükselYiğîter, "The Importance of Logistic Regression Implementations in the Turkish Livestock Sector and Logistic Regression Implementations/Fields",J.Agric. Fac. HR.U., 2012, Vol.16, No.2, 25-36.
[13]ImranNaseem,RobertoTogneri,MohammedBennamoun, "Linear Regression for Face Recognition" IEEE Transactions on Pattern Analysis and Machine Intelligence,Volume: 32, Issue: 11, Nov. 2010.
[14]Priya Stephen, Suresh Jaganathan,"Linear regression for pattern recognition", Green Computing Communication and Electrical Engineering (ICGCCEE), 16 October 2014.
[15]GuoQiang, "Research and improvement for feature selection on naive bayes text classifier", Future Computer and Communication (ICFCC), June 2010.
[16] Sona Taheri, Musa Mammadov, "Learning the Naive Bayes Classifier with Optimization Models", Int. J. Appl. Math. Comput. Sci., 2013, Vol. 23, No. 4, 787–795.
[17] SasanKaramizadeh, Shahidan M. Abdullah, Azizah A. Manaf, MazdakZamani, AlirezaHooman, "An Overview of Principal Component Analysis", Journal of Signal and Information Processing, 2013.
[18] GyotirmoySarkara , SnehanshuSahab , SurbhiAgrawalb, "An Efficient Use of Principal Component Analysis in Workload Characterization-A Study", Conference on Sports Engineering and Computer Science (SECS), 2014.
[19] Victor R. Basili, Fellow, Lionel C. Briand, and Walcelio L. Melo, "A Validation of Object- Oriented Design Metrics as Quality Indicators", IEEE Transaction on Software Engineering, Vol 22, No. 10, October 1996.
[20] Ping Yu, TarjaSysta and Hausi Muller, "Predicting Fault-Proneness using OO Metrics: An Industrial Case Study", Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR.02)1534-5351/02 $17.00 © 2002 IEEE.
[21] 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 Transaction on Softwrae Engineering, Vol. 33, N0. 6, June 2007.