I.J. Education and Management Engineering, **2017, 3, 35-42**
Published Online May 2017 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2017.03.04

Available online at http://www.mecs-press.net/ijeme

An Investigation on the Metric Threshold for Fault- Proneness

Aarti [a], Geeta Sikka [b], Renu Dhir [c]

[a]*CSE Department, Dr B R ambedkar NIT Jalandhar*
[b]*Associate Professor and Head(CSE Department), Dr B R ambedkar NIT Jalandhar*
[c]*Associate Professor, CSE Department, Dr B R ambedkar NIT Jalandhar*

Abstract

The software quality can be enhanced with the awareness and compassionate about the software faults. We acknowledge the impact of threshold of the object-oriented metrics on fault-proneness. The prediction of fault-prone classes in early stage of the life-cycle assures you to allocate the resources effectively. In this paper, we proposed the logistic regression based statistical method and metric threshold to reduce the false alarm for projects that fall outside the risk range. We presented the threshold effects on public datasets collected from the NASA repository and validated the use of threshold on ivy and jedit datasets. The results concluded that proposed methodology achieves the speculative results with projects having similar characteristic.

Index Terms: Quality, semi-supervised, fault- proneness, false-alarm.

1. Introduction

Software quality assurance is becoming more crucial activity and major subset of testing, verification, validation, fault tolerance and prediction [1]. Software testing is time consuming task and sometimes, errors may still left in the software projects even after testing. The crucial parameter to estimate the quality of the projects throughout the life cycle phases are software metrics. These are indicators of the intensity of the complexity in the software which leads to the efficient testing and maintenance [4]. The metrics enables the testers to measures the quality of software system and metrics should be validated with the previous research [13-16]. Some of the qualitative improvement methods include the code inspection, prototypes evaluation, design walk-through and measurement based analysis. The researchers are interested in identification of problematic area to reduce false alarm to maximize the accuracy [6].

The maintenance of software is one of the important activities in the SDLC and testing is decisive assignment under the software maintenance [11-12]. With the increase in the size of software, it becomes difficult to test every part of software. In inclusion to that, exercising the testing to all parts of software is not

* Corresponding author.
E-mail address: aarti.1208@gmail.com

possible. Thus, it is beneficial for the testers to identify on the fault-prone classes to deliver the high quality software. In this paper, we investigate the effect of threshold on fault-proneness where threshold determines the number of classes within the tolerable risk [1-4]. After calculating the threshold, their results are calculated on the OO metrics. The more is the value of metrics above the threshold value; more is the risk of faults in software systems. This allows the testers to modify or redesign the classes to build the qualitative products.

The rest of this paper is organized as follows: Section 2 gives description of data sets and the evaluation criteria opted for quality model. Section 3 discuss about the methodology and modeling technique used for fault proneness with design of experiment. In the end the conclusion is made in Section 5.

2. Data Sets and Metrics

The datasets are taken from PROMISE repository. We have considered the data from NASA aerospace projects and SOFTLAB which is software company dealing with embedded controller applications. Table 1 provides the detailed information about two projects considered in this project with the defect information.

Table 1. Dataset

System	Language	No of classes	% Defects
ivy 2.0	Java	352	11.36
jedit 4.0	Java	306	24.51

The projects considered in Table 1 have various matrices but we have considered those which are common in all analyzed projects [7-8]. The set of selected metrics are depicted in Table 2. The data of software defects have been calculated by ckjm tool along with other software metrics.

Table 2. CK Metrics

Software Metric	Description
CBO(Coupling between Objects)	Two classes are said to be coupled if one class calls method of other class. Inheritance and polymorphism are used in it.
DIT(Depth of inheritance)	Maximum length of class hierarchy that counts the number of ancestor nodes.
LCOM(Lack of cohesion among methods)	Measure degree of dissimilarity of methods in a class along with attributes
NOC(Number of children)	Counting the number of immediate decedents of the class
RFC(Response for a class)	Number of methods that can be executed in response to a message
WMC(Weighted method count)	Summing up of complexity of all methods
SLOC(Source line of code)	Total numbers of lines

The focus of our research is identifying the relationship between OO design metrics and fault proneness. The classification model is constructed with logistic regression based on the threshold value. The ability of fault-proneness model is evaluated on the basis of classification / prediction of projects in fault prone and non fault prone modules. The numbers of researchers have investigated the relationship between the CK metrics and fault-proneness. But in this, we considered the relationship between the CK metrics and fault-proneness based on the threshold values[5]. We also explored that OO metrics were more successful in acquisition of faults than other metrics [10]. In OO metrics, CK were most used among researchers and we analyzed that some of metrics were more effective in predicting the fault proneness and some are not.

2.1. Evaluation Parameter

We will use the commonly used performance measures: accuracy, recall, specificity and precision to evaluate the prediction. The first metric we used is Precision. It is given by the Eq. 1.

$$Precision = \frac{TP}{FP+TP} \tag{1}$$

The second metric to consider is Recall (also called probability of detection, PD). It is given by Eq. 5.

$$Recall = \frac{TP}{FN+TP} \tag{2}$$

The third metric is Probability of false alarms PF. It is calculated as Eq. 6.

$$Probaility\ of\ false\ alarm(PF) = \frac{FP}{TN + FP}. \tag{3}$$

The fourth metric included is accuracy which defines as proportion of predicted fault-prone fault that are inspected out of all module given in Eq. 4

$$Accuracy = \frac{TN+TP}{TP+TN+FP+FN} \tag{4}$$

The fifth metric is the Specificity. This is calculated according to Eq. 5

$$Specificity = \frac{TN}{FP+TN} \tag{5}$$

3. Prerequisite Knowledge

3.1. Threshold Deviation

In this research, we extract metrics based on the significance with fault-proneness. For classification, threshold value is calculated based on mean and standard deviation [9]. In this research, concept is conducted on the ivy-2.0 and jedit 4.0 datasets. We use following calculation to calculate the threshold:

$$T = Mean(\mu) + Standard\ deviation(SD) \tag{6}$$

Now this threshold value (cut-off value) can be used for classification. If the metrics results are founded above the threshold value, then it is considered as fault-prone; otherwise non fault-prone. Logistic regression is used for classification of faults with the cut off value extracted from Eq 6. In simple regression, response variable is linear function of coefficients.

$$Y = B_0 + B_1 * X_1 \dots\dots..+B_n * X_n \tag{7}$$

Regression analysis is statistical technique to find the relationship among OO metrics. But for dichotomous variable, we set up linear model to predict the individuals if response variable represents the two variable. It is also used to interpret which among the independent (OO metrics) are related to the dependent variable.

Table 3 gives the descriptive statics of ivy dataset. Following parameters are used to evaluate the accuracy of model:

B: It is coefficient to the constant value

S.E. It is the standard error around the constant value and used for testing parameter whether it is significantly different from 0.

Statistical significance: It measures the significance levels of the coefficients of attributes measured using logistic regression. The higher is value of significance; lower the estimated impact of independent attribute.

Table 3. Descriptive Statistics of Ivy 2.0

	Coefficients		**Sig.**	**Mean**	**Median**	**Mode**	**S.D.**
	B	**S.E.**					
(Constant)	-.023	.047	.624				
wmc	.006	.004	.115	11.28	6.00	3.00	15.1
dit	-.018	.019	**.329**	1.79	1.00	1.00	1.24
noc	-.011	.017	**.509**	0.37	0.00	0.00	1.32
cbo	.002	.002	.280	13.23	8.00	6.00	16.5
rfc	.001	.002	**.795**	34.04	19.00	2.00	44.6
lcom	.000	.000	.002	131.58	6.00	0.00	712.1
loc	.001	.000	.007	249.34	85.50	1.00	428.3

In table 3, DIT, NOC and RFC are considered as significant, as cut off value taken as 0.3. Table 3 also explored the univariate analysis of CK-metric with mean, median, mode and standard deviation as parameters. Some of the metrics have negative B value means that larger the value of metric have higher impact on the prediction strategy.

Table 4 explored the descriptive statics of jedit 4.0. In which, WMC, NOC and CBO are considered as significant as cut off value is 0.3. Std. Error indicates that number of observed values falls below the regression line. On the other side, it also discusses about how wrongly regression model uses entities of response variable. The standard error is lower for LCOM and LOC and highest standard error found for NOC. **Sig.** indicates the statistical significance of the regression model. If p value is less than 0.06, then we considered regression model as statistically significant and if it greater than 0.06, then it is considered as non-statistically significant. The quality prediction are said to be successful if they meet criteria of less error rate. The highest value of S.E. is found for the CBO. From value of S.E., we found that this metric cannot provide much contribution.

Table 4. Descriptive Statistics of Jedit-4-0

	Coefficients		**Sig.**	**Mean**	**Median**	**Mode**	**S.D.**
	B	**S.E.**					
wmc	-.022	.028	**.417**	12.88	6.00	2	30.958
dit	-.095	.081	.239	2.76	2.00	1	2.119
noc	-.072	.108	**.504**	.44	0.00	0	2.699
cbo	.010	.017	**.574**	12.40	7.00	4	18.041
rfc	.042	.009	.000	38.24	22.00	2	57.019
lcom	.000	.000	.103	197.38	4.00	0	1221.246
loc	-.001	.000	.244	473.21	172.00	5	1584.691

4. Result Analysis

We use ivy 2.0 and jedit-4.0 dataset for prediction of fault-proneness classes. Firstly CK extracted metric are extracted and then, threshold value is calculated. But, this value is very large. To apply logistic regression, this value is normalized between 0 and 1. For normalization, Eq 8 is considered.

$$T_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (8)$$

Following steps are needed for classification using logistic regression:

1. CK metrics are extracted from ivy 2.0 and jedit 4.0
2. Calculate the threshold using Eq. 6
3. Normalized threshold value between 0 and 1 using Eq 8.
4. Normalized threshold is used as cut-off value for the logistic regression.
5. Classify the model using logistic regression.

After extracting the CK metrics, threshold for both projects are calculated. Table 5 gives the original threshold and normalized threshold values for ivy 2.0.

Table 5. Threshold Values for Ivy 2.0

	wmc	**dit**	**noc**	**cbo**	**rfc**	**lcom**	**loc**
Ivy 2.0	233.79	37.08	7.76	274.09	705.1	2820	5173.66
Normalized_ Value	0.06	0.06	0.06	0.06	0.06	0.06	0.06

DIT Threshold: Larger DIT leads to more complex classes and it is used to mark classes that need more attention during testing and maintenance phase.

NOC threshold: It is indicator of both inheritance and abstraction in the classes. Larger is the value, more effort for testing and maintenance phases.

CBO threshold: Higher is the coupling, more complex are classes.

RFC threshold: The classes with higher RFC are more fault-prone and require more maintenance.

LCOM threshold: Low cohesive classes are less structured and difficult to maintain.

Table 6. ROC Values for Ivy 2.0

	precision	**recall**	**pf**	**accuracy**	**specificity**	**sensitivity**
wmc	0.11	0.98	1.00	0.12	0.00	0.98
dit	0.12	1.00	1.00	0.12	0.00	1.00
noc	0.12	1.00	1.00	0.12	0.00	1.00
cbo	0.14	0.45	0.95	0.15	0.05	0.45
rfc	0.19	0.90	0.50	0.55	0.50	0.90
lcom	0.07	0.51	0.86	0.18	0.14	0.51
loc	0.18	0.90	0.55	0.50	0.45	0.90

Table 6 explored the Receiver operative curve (ROC) parameters using univariate logistic regression with threshold value of 0.06 for all metrics. The highest value of precision found for RFC and LOC metrics. Higher is precision, lesser is the probability of false alarm. RFC also gives the higher sensitivity than other metrics.

Table 7 gives the threshold and normalized threshold value for Jedit 4.0 for all the metrics. All the metrics have different normalized threshold value. This threshold is considered as cut-off value in logistic regression for classification. This cut-off value can classifies the projects in true-positive, true-negative, false-positive and false-negative classes. The classification helps us to identify the different ROC parameters.

Table 7. Threshold Value for Jedit 4.0

	wmc	dit	noc	cbo	rfc	lcom	loc
Ivy 2.0	43.8	4.87	3.14	30.4	95.1	1417.9	2056.4
Normalized_ Value	0.17	0.55	0.06	0.16	0.19	0.09	0.23

Table 8 gives the precision, recall, pf, accuracy, specificity and sensitivity values. For classification model, pf should be lower and precision should be higher. Among all the metrics, RFC found to be significant than others. We found comparable results for all the metrics as recall value is approximately same except LOC. DIT is significant among all the metrics as pf value is lower among all.

Table 8. ROC Values for Jedit 4.0

	precision	recall	pf	accuracy	specificity	sensitivity
wmc	0.20	0.67	0.87	0.26	0.13	0.67
dit	0.45	0.07	0.03	0.75	0.97	0.07
noc	0.18	0.67	0.99	0.17	0.01	0.67
cbo	0.18	0.67	0.97	0.19	0.03	0.67
rfc	0.40	0.79	0.39	0.65	0.61	0.79
lcom	0.25	0.67	0.65	0.43	0.35	0.67
loc	0.43	0.99	0.43	0.67	0.57	0.99

5. Conclusion

We investigated that the object-oriented metrics plays most important role to predict the fault-proneness in software projects. The achievement of software quality is one of the major issues. There is appropriate tool and methods are needed to identify fault-prone classes. To improve error prediction accuracy, we analyzed quality metrics using threshold technique to identify the complex classes. In our research two methods are employed: threshold calculation and prediction using logistic regression with the calculated threshold value for each metric.

References

[1] V. Basili, L. Briand L and W. Melo, A validation of object-oriented design metrics as quality indicators. IEEE Transactions on Software Engineering, vol. 22, no. 10: 1996, pp. 751–761

[2] K. Ferreira, M. Bigonha, S. Bigonha , L. Mendes and H. Almeida , Identifying thresholds for object–oriented software metrics, Journal of Systems and Software, vol. 85, no.2, 2012, pp. 244–257.

[3] C. Catal, U. Sevim, and B. Diri, B. Clustering and metrics threshold based software fault prediction of unlabeled program modules, In proceeding of Sixth International Conference on Information Technology: New Generations, 2009, pp. 199-204.

[4] C. Catal, Software fault prediction: A literature review and current trends, Expert Systems with Applications, vol. 38, no. 4, 2011, pp. 4626-4636.

[5] S. Chidamber and C. Kemerer, A Metrics Suite for Object-Oriented Design, IEEE Transaction of Software Engineering, vol. 20, no.6, 1994, pp. 476-493.
[6] T. Hall, S. Beecham, D. Bowes, D. Gray and S. Counsell, A systematic literature review on fault prediction performance in software engineering, IEEE Transaction Software Engineering, vol. 38, no.6, 2012, pp. 1276-1304.
[7] P. K. Rajput, G. Nagpal and Aarti, Feature weighted unsupervised classification algorithm and adaptation for software cost estimation, Intertaional Journal of Computational Intelligence Studies, vol. 3, no. 1, 2014, pp 74-93.
[8] P. K. Rajput , G. Nagpal and Aarti , CGANN-Clustered Genetic Algorithm with Neural Network for Software Cost Estimation, In proceeding of International Conference on Advances in Engineering and Technology (ICAET'2014), March 2014, pp 268-272
[9] R. Perez-Castillo, L. Sánchez-González, M. Piattini, F. Garcia , I. Garcia-Rodriguez, Obtaining thresholds for the effectiveness of business process mining, In proceedings of the International Symposium on Empirical Software Engineering and Measurement, 2011, 453–462
[10] T. Gyimothy , R. Ferenc, I. Siket, Empirical validation of object-oriented metrics on open source software for fault prediction, IEEE Transactions on Software Engineering,vol. 31, no. 10, 2005, pp.897–910.
[11] J.M. Bieman and B.K. Kang ,Cohesion and Reuse in an Object-Oriented System. Proceedings of the Symposium on Software reusability-ACM 20: 1995, pp. 259-262
[12] S. Bhattacharya , S. Rungta and N. Kar N, Software fault prediction using fuzzy clustering & genetic algorithm. International Journal of Digital Application & Contemporary Research, vol. 2, no. 5, 2013, pp. 1-7
[13] P. K. Rajput, G. Nagpal and Aarti, Feature weighted unsupervised classification algorithm and adaptation for software cost estimation. International Journal of Computational Intelligence Studies, vol.3, no. 1, 2014, pp. 74-93.
[14] P. K. Rajput, G. Nagpal and Aarti, CGANN-Clustered Genetic Algorithm with Neural Network for Software Cost Estimation. In proceeding of International Conference on Advances in Engineering and Technology (ICAET'2014), 2014, pp. 268-272.
[15] Aarti, G. Sikka, R. Dhir, An investigation on the effect of cross project data for prediction accuracy. International Journal of System Assurance Engineering and Management, vol. 7, no. 1, 2016, pp. 1-26.
[16] A. H. Patil, N. Goveas and K. Rangarajan, Regression Test Suite Execution Time Analysis using Statistical Techniques. I.J. Education and Management Engineering, vol. 6, no. 3, 2016, pp. 33-41.

Authors' Profiles

Aarti received her Master of Technology in Computer Science and Engineering from NIT Jalandhar and she is currently pursuing her Ph.D. degree from NIT jalandhar. She is also serving as assistant professor at DAV University. Her areas of interest are artificial intelligence, genetic algorithm, data mining and cryptography.

Dr. Geeta Sikka has done her Ph.D. from Dr. B.R. Ambedkar NIT, Jalandhar. She received her M-Tech in Computer Science and Engineering from Punjab Agricultural University, Ludhiana, India. She is currently working as Head of computer science and engineering department at Dr B R Ambedkar NIT, Jalandhar. Her areas of interest include data mining, software engineering and artificial intelligence. She has published many papers in national and international conferences and journals.

Dr. Renu Dhir has done Ph.D. from Punjabi University in 2008 and completed her M.Tech. (Computer Science & Engineering) in 1997 from TIET, patiala. She is currently working as associate professor in computer science and engineering department at Dr B R Ambedkar NIT, Jalandhar. Her fields of research are mainly character recognition, pattern recognition and image processing and she has published more than 35 papers in national and international conferences and journals.

How to cite this paper: Aarti, Geeta Sikka, Renu Dhir,"An Investigation on the Metric Threshold for Fault-Proneness", International Journal of Education and Management Engineering(IJEME), Vol.7, No.3, pp. 35-42, 2017.DOI: 10.5815/ijeme.2017.03.04