Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Scalability and Sensitivity of WCS and WChS over the Traditional Coupling and Cohesion Metrics: An Empirical Analysis

Amjan Shaik*
Research Scholar, Dept.of CSE
JNTUH, Kukatpally
Hyderabad
Andhra Pradesh, India

Dr.C.R.K Reddy
Professor & HOD-CSE
CBIT, Gandipet
Hyderabad
Andhra Pradesh, India

Dr.A.Damodaram
Professor of CSE, Chairman BOS
Director AAC, JNTUH, Kukatpally
Hyderabad
Andhra Pradesh, India

Abstract— In this Research Paper the experimental study of WCS and WChS metrics performed in detail by evaluating with other regularly mentioned methods. CBO, LCOM and RFC are used for evaluating. This article also shows an experimental methodology of importance of Weighted Coupling Support (WCS) and Weighted Cohesion Support (WChS) also with design metrics like Response for a Class (RFC), Coupling Between Object classes (CBO), Number of Children (NOC), Depth of Inheritance Tree (DIT), Lack of Cohesion Between Methods (LCOM), Weighted Methods per Class (WMC) in defect prediction models. Several releases of various open-source projects were checked upon. Each metrics was checked upon with the Pearson correlation coefficients with amount of defects. Later defect prediction models were developed making use of linear stepwise regression and discriminatory analysis was performed. As the stepwise regression was made use, the acquire models always had a subset of the investigated metrics.

Keywords— Empirical Analysis, WCS, WChS, Logistic Regression, Linear Regression

I. INTRODUCTION

Necessity for productive software has been culminating and OO design technique is providing solution to this as it is the most powerful mechanism for developing proficient software systems. It is helpful in price decrease and for development of high quality software systems. Software developers require accurate metrics for developing efficient software system. Object Oriented Metrics (OOM) plays a significant role pertaining to this aspect because of their importance in the development of successful software applications. Concepts such as complexity, usability, reusability, testability, understandability etc. are utilized for improving the quality of software system. Software metrics is now very essential for many areas of software engineering because it is employed for calculating the software excellence along with calculating the expenditure plus endeavor of software projects.

II. DESCRIPTION OF DATA AND METRIC SUITE

Data set used for the study is presented in this section. A small foreword to the metric suite if also shown here. Hence it was feasible to know if the selected metrics relates to the correlation with the amount of defects and the classified power of the metrics. Also as per the results got, the importance of WCS and WChS metrics in defect forecasting are high and couple of metrics were suggested along with WCS and WChS in respect to defect prediction. Regression analysis methods were applied. These are popularly used to calculate an unknown variable based on one or more known variables. Logistic regression [12] was chosen to develop the relationship between the metrics and fault-prediction. The classes were categorized as faulty and non-faulty. Linear regression [11] was applied to know the number of defects. Both univariate and multivariate models were done using these methods. These procedures were applied by popular researcher for related kind of predictions [2], [3]. Handing over and checking the distribution of measures is mandatory for the evaluation of various metrics used. Additional analysis makes use of more than six non-zero data points that are measured for more analysis. The Table 1 shows descriptive statistics.

A. `Dataset

This study makes use of an open source project analysis logs from source forge. The data assembled was of a search engine API and was applied in the java programming language. It has 145 classes which include 2107 methods and sum total of 40k lines of code. There were 60 faulty classes with the rest of classes which did not have any fault.

B. Metric Suite and Fault Proneness

Table 1 shows the various ways of predictor software metrics (independent variables) that are made us of in the study. It is common to check big numbers of measures since they tend to be investigative for studies. The study uses static OO measures [9] that were collected during the development process. All measured as independent variables are made use of,

which are measures of class size, inheritance, coupling and cohesion. These are at class levels. Popular metrics like McCabe's Cyclomatic difficulty and executable lines of code come under the traditional static difficult and size metrics. These are generally available at the site for technique level but are later changed to class level and their common values are made use of for the studies. Henry and Kafura [8] given by another traditional metric is also made use of.

Table1: Metric suite at class level used for the study

Metric	Definition
Coupling Between Objects (CBO)	The number of distinct non-inheritance-related classes on which a class depends.
Depth of Inheritance (DIT)	The level for a class. Depth point out at what level a class is located in its class hierarchy.
Lack of Cohesion between Methods (LCOM)	It counts number of null pairs of methods that do not have common attributes.
Fan in (FANIN)	This is a count of calls by higher modules.
Response For Class (RFC)	A count of methods implemented within a class plus the number of techniques accessible to an object class due to inheritance.
Weighted Methods Per Class (WMPC)	A count of techniques implemented within a class (rather than all techniques accessible within the class hierarchy).
Number Of Children (NOC)	It is the number of classes derived from a specified class.
Cyclomatic Complexity (CC)	It is a measure of the complexity of a modules decision structure. It is the number of linearly Independent paths.
Weighted Cohesion Support (WChS)	Weighted Cohesion Support of the classes that measured by a statistical measuring approach.
Weighted Coupling Support (WCS)	Weighted Coupling Support of the classes that measured by a statistical measuring approach.

This is made use of to measure interclass coupling by checking the quantity of calls passed to the class. Also there is the computed metric which is known as the techniques per class. This checks the quantity of modules per class. The original detail here is that it will be more fault prone as the number of techniques per class will increase. To check whether the module is fault prone or not the dependent variable makes use of the Defect level metric. The dependent variable or the response variable gets a true response if the class has one or more defects and gets a false if there are no defects. The variable will have a numeric value if it is false and 0 if it is true. In linear regression modelling the reliant variable is nothing but the genuine number of faults in the class. The connection between the metrics and the fault proneness can be arrived as follows:

- The length or the size of the metrics can be got from the size related metrics. The class is said to be fault prone if its size is said to be bigger.
- DIT, NOC are inheritance related metrics that help in arriving the amount of potential reuse of the class. Higher up the reuse in the class more will be it difficult and fault prone.
- The co-dependence between every components of the software system can be measured by coupling metrics. System can get difficult if the coupling is high. The fault proneness will be more if the coupling is more.
- The intricacy metrics of a class tells the complexity of a modules decision structure. The class will have more faults as the difficulty in the class is more.
- Class cohesion is stated by the cohesion metrics and classes with less cohesion than its peers os said to be more fault prone.

C. Defect Analysis

Here the analysis done to know the relationship between different metrics and the number of defects detected in each class. Regression analysis methods were used largely to foresee an unknown variable based on one or more acknowledged variables. The relationship between metrics and fault-prediction can be studied using logistic regression [7]. These categorize the classes as faulty and non-faulty. Linear regression [6] was applied to know the number of defects. There are many univariate and multivariate models that use both the methods. Several known researchers have applied these methods for same kind of predictions [1].

III. RESEARCH METHODOLOGY AND ANALYSIS RESULTS

Here the analysis done to know the relationship between different metrics and the number of defects detected in each class. Regression analysis methods were used largely to foresee an unknown variable based on one or more acknowledged variables. The relationship between metrics and fault-prediction can be studied using logistic regression

[7]. These categorize the classes as faulty and non-faulty. Linear regression [6] was applied to know the number of defects. There are many univariate and multi-variate models that use both the methods. Several known researchers have applied these methods for same kind of predictions [1].

A. Data Analysis

The distribution max, mean, standard divergence and difference of every independent variable is checked in the case study. Minor difference measures do not differentiate the classes perfectly and hence are not of much help. Showing and knowing the distribution of measures is important for assessing the various metrics used. Those measured with more that sis non-zero data points are taken up for further analysis.

Table 2 shows the significant statistics given below. Around 145 classes are present for which all metrics are available. NOC and FANIN are not important for the results as they have low variance. As various measures are captured by each metrics all the measures are used in the modelling.

Table 2: Descriptive Statistics of the data used

	CBO	DIT	LCOM	FANIN	RFC	WMPC	NOC	CC	WChS	WCS
Mean	8.32	2.00	68.72	0.63	34.38	17.42	0.21	2.46	11.83	14.53
Median	8.00	2.00	84.00	1.00	28.00	12.00	0.00	1.86	8.67	10.00
Std.Dev	6.38	1.26	36.89	0.69	36.20	17.45	0.69	1.73	12.12	15.26
Variance	40.66	1.58	1360.77	0.48	1310.65	304.47	0.49	2.99	146.88	232.99

B. Logistic Regression Analysis

The data which has Dependent Variable (DV) has Logistic Regression (LR) [7] applied to it. The DV is of dichotomous nature that is it may be present or absent. This is used to calculate the chances of an event happening, for example fault detection. Unlike Linear Regression LR does not understand any severe functional form to link clarifying variables and the probability function. LR works on probability and presumes that all observations are not dependent. To know which data points are influential outlier analysis is done and eliminating them is also important. To distinguish multivariate outlier we analyze each data point the Mahalanob is Jackknife distance. In [5] the details of this outlier can be calculated.

The following equation defines the Multivariate Logistic Regression (MLR) model. It is a univariate model if it has only one independent variable.

$$\frac{\pi(X_1, X_2 .... X_N)}{1-\pi} = \frac{Prob(event)}{Prob(nonevent)} = e^{B_0 + B_1 X_1 + ... + B_N X_N} \quad \text{Equation – 1}$$

Where

X_i '' s are the independent variables and is the probability of occurrence of a fault.

B_i ' s are the probable regression coefficients of the LR equation. The extent of impact is shown by all descriptive variable on the predictable probability and hence the importance of every explanatory variable. The crash of the explanatory variable on the predictability of the fault being detected in a class is strong if the complete value of the coefficient is bigger.

MLR is performed to construct a precise prediction model for the variable that is dependent. The relationships between the Independent Variables (IVs) and the Dependent Variable (DV) is looked upon but only the previous in combination as covariates in the multivariate model is considered to get the exact predictions. To check the correctness of the predictions, various modelling methods that have exact measures of good are used. In the existing study subsequent statistics are made use of to reveal and check the new results obtained.

Redundant data may be captured by the examined metrics and they are also found to be dependent. The preciseness of these studies are said to be explanatory in nature and also there is no apt theory which can suggest the variables that could be included in this prediction model and which could be left out. In such a situation the selection process can be stepwise in the prediction models. Each step will have one variable either entering or leaving the model. The common backward removal process commences with a model that has all independent variables. Till the close criterion is reached the variables will be selected one at a time to removed from the model.

Table 3: Univariate Logistic Regression model

COVARIATES	CBO	DIT	LCOM	FANIN	RFC	WMPC	NOC	CC	WChS	WCS
Coefficient	0.207	0.054	0.000	0.351	0.010	0.025	-0432	0.4	0.076	0.076
Constant	-2.167	-0.456	-0.343	-0.575	-0.679	-0.789	-0.269	-1.336	-1.256	-1.384
P(sig)	0.000	0.687	0.987	0.153	0.060	0.030	0.184	0.001	0.000	0.000
R2	0.263	0.001	0.000	0.014	0.027	0.040	0.016	0.093	0.147	0.140

Where

P is the statistical importance of the LR coefficients, gives an insight into the exactness of the coefficients estimates. A significance threshold of (i.e.,5% probability) has regularly been used to find whether a variable is a essential predictor.

(R-square) coefficient is the fairness of fit, not to be puzzled with least square regression. Both are built upon different formulae, even though they both range between 0 and 1.The higher, the higher the effect of the model explanatory variables, the more exact the model. Though this value is often low for LR. Cox and Snell's R-Square in the present study, which is a replication to the understanding of multiple R-Square for binary LR.

MLR model is also checked for multicollinearity. If are the covariates of the said model then the chief part study on these variables gives to be the biggest Eigen value the negligible Eigen value of the main components. The conditional number is then defined as. A large conditional number (i.e., difference among minimum and maximum eigenvalue) indicate the happening of multicollinearity and should be under 30 for appropriate limits [13]. The model is made by all the 145 classes for the training. The model has various metrics as shown in univariate and multivariate models exposed in Table 3 and 4. DIT, LCOM, FANIN, and NOC are the metrics that are shown as insignificant in the univariate model. NOC, CC and FANIN are not present in the multivariate model. A classifier at threshold = 0.5 is used to categorize the classes as accessible in Table 4. The test for multicollinearity for the model is performed by calculating the provisional number is that is well below 30. Apart from this no vital outlier was seen in the model. The correctness of the model is 79.31%, recall is 71.67% and the accuracy is 76.79%.

This model is also checked for accuracy in terms of precision and totality. Two regular measures are made use for classifications. Exactness is the number of classes correctly shown as fault prone, divided by the complete number of classes termed as fault-prone. Enhanced precision indicates that effort in fault detection will be smaller. Therefore progress in the competency can be achieved. 76.79 % is the present correctness. Parameter fullness [10] is said to be the number of faults in classes termed as fault prone, divided by the total number of faults in the system. Larger the fullness in the value more will be the number of errors will be found in the faulty predicted classes. In the existing study the comprehensives is 568 out of 669 which is 84.90 percent.

Table 4: Multivariate Logistic Regression model

COVARIATES	CBO	DIT	LCOM	RFC	WMPC	WChS	WCS	CONST.
Coefficient	0.23	-1.3	-0.02	0.04	-0.24	-0.07	0.26	-0.92
P(sig)	0.001	0.002	0.028	0.091	0.079	0.021	0.061	0.150
-2 Log likelihood	118.792							
R^2	0.416							

C. Linear Regression Analysis

For classification problems it is best to make use logistic regression. In the existing study, there are several classes that have unpredictable number of defects. This situation can be outlined using linear regression [4]. Explanatory variables are the consistent metrics and dependent variable will tell you the amount of errors in a class in linear regression. The precision of the model can be checked by the rightness of the fit of the model and the statistical significance of the parameters. Two important test statistics that are made use of are the multiple coefficients of determination R^2 and the t-value. R^2 is the disparity percent on the variable, which is dependent and is accounted by the regression model. Regression coefficients are made more equivalent when standardized coefficients or betas are used

with regularly independent variables as measures in different units. Table 5 and 6 shows the results of univariate and multivariate models. DIT, LCOM, FANIN and NOC come in again as insignificant predictors for modeling the number of defects. WChS and DIT are both present in the multivariate model while CC makes an astonishing entry. The value of R^2 is 0.422 that shows that 58 percent of the variance is not made clear by the model. The other dissimilarity that are not taken into consideration are external environmental and psychological factors. These are not taken into account for the said metric set. There can be many other reasons for it.

Table 5: Univariate Linear Regression model

COVARIATES	CBO	DIT	LCOM	FANIN	RFC	WMPC	NOC	CC	WChS	WCS
Coefficient	0.623	0.320	.037	1.573	0.082	0.288	-1.635	2.057	0.288	0.417
Constant	-0.565	3.973	2.062	3.613	1.807	-0.409	4.963	-0.438	1.203	-1.440
P(sig)	0.000	0.658	0.131	0.228	0.001	0.000	0.208	0.000	0.000	0.000
R^2	0.134	0.001	0.016	0.010	0.074	0.215	0.011	0.107	0.104	0.343

Table 6: Multivariate Linear Regression model

Covariates	DIT	CC	WCS	Const.
Coefficient	-1.987	2.151	0.387	-2.316
T	-2.961	4.340	8.297	-1.599
Beta	-0.230	0.342	0.544	---
R^2	0.422			
Adjust R^2	0.410			

IV. CONCLUSIONS

The models are all tested on the same information hence the results can be positive. It could further extended as a complete model over the software systems from various environments. The model is created on measures that are collected from previous investigations and design artifacts that are based on the size of the source code. These can be problem solving for the final developed system. Hence the use of known models based on previous artifacts and their skill to calculate the quality of the final system remains to be checked. Moreover datasets must be form comparable environments so that model from the present system measures can be used to other systems that are being made. The conclusions drawn are prejudiced as per the information that is made use of to generate them. For example the sample made use of here is open source projects in a chief sole market niche. Here it can be contested if the results from the open source are related to the common software engineering industry. Fault prediction models are predicted using the applicability of the statistical methods. There are feasible methodologies for the two. A comparison can be drawn between the feasible methodologies and the regression method in knowing software fault prediction. Both the results can be compared.

In future this study could be used to map the fault proneness of the classed with attempt and time needed to rectify them in the developmental and maintenance phase. A study can also be taken to make use of the metrics to know various maintenance metrics.

REFERENCES

[1] C. Briand, J. W. Daly, V. Porter, and J. Wust A, "Comprehensive Empirical Validation of Product Measures for Object-Oriented Systems", Technical Report, ISERN-98-07, 1998.

[2] L. C. Briand, J. W. Daly and J. Wust, "A Unified Framework for Coupling Measurement in Object-Oriented Systems", IEEE Transactions on Software Engineering, Vol. 25, No. 1, pp. 91–121, 1999.

[3] Fenton, N.E., Neil, M, "A critique of software defect prediction models", IEEE Transactions On Software Engineering, 25(5), pp.675–689 ,1999.

[4] Gyimothy, T., Ference, R., Siket., I, "Empirical Validation of Object –Oriented Metricson open source software for fault prediction", IEEE Transactions on Software Engineering, 31(10), October 2005.

[5] Barnett, V., Price, T, "Outliers in Statistical Data", John Wiley & Sons, Chichester , 1995.

[6] Neter, J., Wasserman, W., Kutner, M.H, "Applied Linear Statistical Models", 3rd Edition. Richard D.Irwin ,1990.

[7] Hosmer, D., Lemeshow, S, "Applied Logistic Regression", Wiley-Interscience, Chichester, 1989.

[8] Henry, S., Kafura, D, "Software structure metrics based on information flow", IEEE Transactions on Software Engineering, 7(5), pp.510–518, 1981.

[9] Chidamber, S.R., Kemerer, C.F, "A Metrics Suite for Object Oriented Design", IEEE Transactions on Software Engineering 20(6), pp.476–493, 1994.

[10] Briand, L.C., W¨ust, J., Daly, J.W., "Assessing the applicability of fault-proneness models across Object-Oriented software projects", IEEE Transactions on Software engineering 28(7),706–720, 2002.
[11] D.Tegarden, S. Sheetz, D.Monarchi , "A Software Complexity Model of Object Oriented Systems, Decision Support Systems", vol. 13 no.3-4, pp.241-262, 1995.
[12] http://msr.uwaterloo.ca
[13] Belsley, D., Kuh, E., Welsch, R. "Regression Diagnostics: Identifying Influential Data and Sources of Collinearity", John Wiley & Sons, Chichester ,1980.

