Journal of Engineering and Applied Sciences 13 (1): 7-10, 2018
ISSN: 1816-949X

Validation of Object-Oriented Software GA Metric Selection Model using Domain Experts

[1, 2]Abubakar Diwani, [1]Abu Bakar Md Sultan, [1]Hazura Zulzalil and [1]Jamilah Din
[1]Faculty of Computer Science, University Putra Malaysia (UPM),
43400 Serdang, Malaysia
[2]Department of Computer Science, The State University of Zanzibar,
P.O. Box 146, Zanzibar, Tanzania

Abstract: This study presents validation of object-oriented model to predict its maintainability. The study used metric threshold in its encoding strategy in the implementation of GA Model before being compared with classical model. This empirical validation was then compared with real maintainability data from experts using similar procedures. To understand the overall effect of particular software, linear discriminant analysis which is machine learning statistical method was utilised to evaluate the performance of the metrics. The results pointed out that there is significant relationship when expert's opinions were used. Experts also indicated the role of inheritance metrics in predicting maintainability of object-oriented software which also highlighted the needs for further empirical investigation on the production of more metrics threshold that give researchers and practitioners an opportunity to work on more metrics.

Key words: Maintainability, metric threshold, expert's opinions, linear discriminant analysis, statistical, validation

INTRODUCTION

For many software development researches, maintainability has dominated the field as is a major quality criteria for both technical and managerial (Dagpinar and Jahnke, 2003; Mizuno and Hata, 2010). For example to understand maintenance efforts required for certain software, software metrics have been used to enumerate the attributes of that software (Preece, 2001). Practice shows that both single software metric and the suite (group of software metrics) has been used.

Although, several quality models have been proposed in predicting different software quality attributes like maintainability but they face difficulties in their validation process (Emam, 2002; Pai and Dugan, 2007). The aim of this study is to validate the GA Software metric selection model in predicting object oriented software maintainability. The model used software metrics thresholds to encode the chromosomes (parameters in GA problem). The classified metrics from GA Model were primarily comparing with two traditional classification model (CK suite and PCA) in two different cases (Geotool and Geoserver), GA results was promising. In this supplement study, the proposed model used real maintainability data from experts and the GA output showed remarkable performance over Principal Component Analysis (PCA) and software Chidamber and Kemerer (CK) metrics suite.

Expert's opinions model development: In software engineering, human involvement in model development has been discussed by several practitioners. Expert's opinions has been used in development and validation of the models and showed noteworthy results (Adomavicius and Tuzhilin, 2001). For example, Li and Smidts (2003) used experts in their model to rank the best reliability metrics. In this study, experts have given remarkable decision in the validation of the GA metrics selection model to predict software maintainability.

MATERIALS AND METHODS

The study used Linear Discriminant Analysis (LDA) to find linear combination which can separate more than one features. Software metrics selection is characterized as the classification problem where metrics needed to measure the quality of software supposed to be more than one while there are dozens of software available. So, the

Corresponding Author: Abubakar Diwani, Faculty of Computer Science, University Putra Malaysia (UPM), 43400 Serdang, Malaysia

question here is which metrics are capable for particular software. Precision and recall categorization has been used to find the performance of the Chidamber and Kemmerer (CK) metrics. Calculation of precision and recall were based on information generated from the metric values that were then encoded to binary digits presenting complex classes (difficult to maintain) and less complicated ones (easy to maintain). In our case, manipulation is based on the two selection probability of correct information retrieved from the metrics categories based on software metrics threshold. The classification accuracy is between 0 and 1 with the expression:

$$\text{F-measure} = \frac{2\times\text{Recall}\times\text{Percision}}{\text{Recall}+\text{Percision}}$$

Where:

Recall = TP/TP+FN
Percision = TP/FP+TP

The content is the values of the maintainable classes associated with metrics selected and the complex classes related with the selected metrics. Precision and recall for the 6 CK metrics are the probabilities of the less complicated classes and the probability of complicated classes that were then used to calculate the F-measure.

The metrics values obtained using open-source tool called CK Java Metric (CKJM) (Spinellis, 2005). Metrics values generated from the classes of the two geospatial Java Software systems were based on metric thresholds. The Geotool contains 2312 classes and the Geosever with 5530 classes were used in investigating the metrics classification performance.

Expert validation in metric selection model: This study intended to validate software metrics selection model using experts in comparison to other traditional classification technique. The idea is to gain confidence on the developed model. We adopt Kappa score technique to measure the agreement between observers that takes into account the fact that observers will sometimes agree or disagree simply by chance. Normally, the Kappa of 1 indicates perfect agreement whereas a Kappa of 0 indicates agreement equivalent to chance. Table 1 shows Kappa score for two observers.

Dataset consisted of 100 Java classes. Out of the 100 observed classes, all Java interfaces were not taking into consideration. Kappa score (inter-inspector agreement for the three inspectors) was 0.54 (moderate agreement). A difference of zero means identical scores a difference of one indicates the developers were off by one grade for example, an inspector ranked a class 4, another inspector ranked the same class 3 or 5. None of the Java classes were given a completely contradictory ranking; low (rank of 1) by one developer and high (rank of 5) by another.

Table 1: Ranking by inspectors for 100 Java classes

Ranking differences	0	1	2	3	4	Kapa score
Number	85	12	2	1	0	0.54
Percent	85	10	2	1	0	

Table 2: Mapped results by inspectors in three different ranking levels

	Mapped ranking results ranking level		
Variables	Low	Medium	High
Geotool	12	32	56
Geosever	18	35	47

The majority of the agreement 85% of the classes came in labelling the classes as rank 1. Of the rankings with a difference of 2 (12% of the classes), the highest discrepancies were between labels 2 and 3 and between 4 and 5. Thus, inspectors had the most difference when ranking low-medium vs. medium and medium-high vs. high. For the sake of classification with the predictive model, the final rankings were 1-3 (low, medium and high). The original rankings (1-5) were mapped to form new ranking labels low rank if all ranker agree it was ranked 1, a medium rank if 2 or 3 and a high rank if 4 or 5. In this case, for each class, if all participants ranked 1 then new ranking label mapped was to low (1) if participants ranked 2 or 3 then new ranking label mapped to medium (2) and if participants ranked 4 or 5 then new ranking label is high (3). Mapped results by inspectors were arranged into low, medium and high. Table 2 shows the mapped categories.

The mapped result was used to understand the relationship between the expert's opinions on the best model among three techniques. A final comparison of classifier performance based on experts in comparison to genetic algorithm as a search-based metric selection strategy show common results in both cases (Geotool and Geosever). Table 3 illustrates the performances. For the Geotool Software, the whole CK metrics resulted in an F-measure of 0.5698, PCA metrics counts 0.4758 and GA achieved 0.614194. The GA based metrics subset resulted in the best LDA performance compare to other two in case of Geotool. When Geoserver is used, the whole CK metrics resulted in an F-measure of 0.555826, PCA metrics achieved 0.4928 and GA achieved 0.58656. Again in Geoserver GA based metrics subset resulted in the best LDA performance.

Table 3: Performance measurement of Geotool and Geosever software for three different methods based on low, medium and high levels

	Low (12)			Medium (32)			High (56)		
Methods	Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
Geotool									
The whole CK metrics	0.88	0.12	0.2112	0.54	0.32	0.4019	0.58	0.56	0.569825
The PCA metrics	0.88	0.28	0.4248	0.54	0.33	0.4097	0.61	0.39	0.475800
Proposed metrics using GA	0.88	0.12	0.5112	0.54	0.32	0.4019	0.68	0.56	0.614194
Geoserver									
The whole CK metrics	0.82	0.18	0.2952	0.50	0.35	0.4118	0.68	0.47	0.555826
The PCA metrics	0.64	0.36	0.4608	0.80	0.20	0.3200	0.56	0.44	0.492800
Proposed metrics using GA	0.82	0.23	0.5592	0.50	0.30	0.3750	0.78	0.47	0.586560

RESULTS AND DISCUSSION

The model used thresholds as empirical encoding strategy in GA Model in developing metrics classification model. The classified metrics from GA validated in comparison to two classified model (CK suite and PCA) using two cases. In both cases, GA results were promising. Table 4 shows performance for three classification models when applied to both Geotool and Geosever Software when applied to both empirical and experts.

Overall empirical performances for the GA classification performance when applied to the first dataset called Geotool was 0.8293 for the CK suite, 0.8840 for PCA and lead by 0.9199 of GA. In the second dataset called Geoserver, metrics selection performances for the CK suite was 0.7978, the PCA was 0.8898 and the GA was 0.9069. When experts involved, for Geotool Software, F-measure of 0.5698 was observed for the whole CK metrics, PCA metrics was 0.4758 and GA metrics achieved 0.6142. In this part, the GA based metrics subset resulted in the best LDA performance compare to other two cases. For the Geosever, the f-measure of 0.5558 was observed when suite is used, 0.4928 for PCA and GA achieved 0.58656. Again, GA based metrics subset resulted in the best LDA performance.

The overall performance for both objective and subjective and with both cases Geotool and Geoserver, the GA metrics consistently achieved high F-measure for all three levels of complexity.

Threat to validity: The study used Java-based systems as the tools in the experiments. The use of Java-based systems was a strategy taken to be cautious of the misinterpretation found by Ferreira *et al*. (2012). They found that the quality evaluation is always error prone due to the possibility of programming dependent. Since, the geospatial software developed using Java programming, open-source software, the probability of errors is expected to be minor. Therefore, this study also decided to use more than one software product to avoid that threat. This study used a different case from the one used by previous researchers but testing the model in two different products within the same context reduces this menace.

Table 4: Selection performance measures for three different classification techniques

	F-measure using thresholds		F-measure using experts	
Methods	Geotool	Geosever	Geotool	Geosever
The whole CK metrics	0.8293	0.7978	0.5698	0.5558
The PCA metrics	0.8840	0.8898	0.4758	0.4928
Proposed metrics using GA	0.9199	0.9069	0.6142	0.5866

Another threat is the use of some thresholds that were proposed from other quality attributes like reusability. The fact is that there are direct relationships between quality attributes such as maintainability, reliability and reusability. Their relationship is based on internal attributes. Therefore, this threat can also be ignored. In the experts ranking study, 100 classes were used for each software product to represent the complete software product. This is due to the time frame. But the threat was solved by selecting the classes at random in assumption that the programming style in the same product is common.

CONCLUSION

In this study, resaercher present the validation of GA metric selection model in the prediction of object-oriented software maintainability. Real maintainability data from experts were used in order to gain more confidence to the preliminary validation process where result metrics from GA was compared to that from CK suite and PCA. The GA shows promising results and the model can be used by practitioners to identify maintainable software for particular purpose. Finally, the study points out the need for the researchers and other practitioners to produce and to validate more software metric with their thresholds to give practitioners opportunities to work with more metrics.

REFERENCES

Adomavicius, G. and A. Tuzhilin, 2001. Using data mining methods to build customer pro?les. Computer, 34: 74-82.

Dagpinar, M. and J.H. Jahnke, 2003. Predicting maintainability with object-oriented metrics-an empirical comparison. Proceedings of the 10th Working Conference on Reverse Engineering (WCRE), November 13-16, 2003, IEEE, British Columbia, Canada, pp: 155-164.

Emam, K.E., 2002. Object Oriented Metrics: A Review of Theory and Practice. In: Advances in Software Engineering, Comprehension, Evaluation and Evolution, Erdogmus, H. and O. Tanir (Eds.). Springer, New York, pp: 23-50.

Ferreira, K.A.M., A.S.M. Bigonha, R.S. Bigonha, L.F.O. Mendes and H.C. Almeida, 2012. Identifying thresholds for object-oriented software metrics. J. Syst. Soft., 85: 244-257.

Li, M. and C.S. Smidts, 2003. A ranking of software engineering measures based on experts opinion. IEEE Trans. Software Eng., 29: 811-824.

Mizuno, O. and H. Hata, 2010. An empirical comparison of fault-prone module detection approaches: Complexity metrics and text feature metrics. Proceedings of the 34th Annual Computer Software and Applications Conference, July 19-23, 2010, Seoul, South Korea, pp: 248-249.

Pai, G.J. and J.B. Dugan, 2007. Empirical analysis of software fault content and fault proneness using bayesian methods. IEEE Trans. Software Eng., 33: 675-686.

Preece, A., 2001. Evaluating Verification and Validation Methods in Knowledge Engineering. In: Industrial Knowledge Management, Roy, R. (Ed.). Springer, London, pp: 91-104.

Spinellis, D., 2005. Tool writing: A forgotten art? IEEE Software, 22: 9-11.