*Regular Paper*

# On some Fault Prediction Issues

CAMARGO CRUZ Ana Erika[†1] and IIDA Hajimu [†1]

In this paper, we argue that in the field of fault prediction, we may be re-inventing the wheel at using product metrics as predictors that are inter-correlated. We think that the analysis of other kind of metrics is needed, which has not been possible mainly due to the lack of publicly available data using industry settings for research.

## 1. Introduction

During the NATO Software Engineering conference in 1968, Mr. J. Nash from the IBM UK Laboratories suggested test planning and status control as valuable tools in managing the later part of a software development cycle. But, what the research community has done about? Among the various research areas for software testing is fault prediction, which main motivation is to predict faulty components to provide more effective criteria for the elaboration of test cases. A large number of prediction models have been proposed in the last decades, but hardly put into practice.

The most frequent metrics used as predictors are product metrics (e.g. size and design complexity OO). More recently other metrics tailored from logs in code repositories have been studied, such as past number of defects and added/deleted LOC of two different file versions. Recent literature review 1) found LOC to be useful in fault prediction overall. Being LOC one of the first metrics suggested as predictors of faulty code by Akiyama et al. in 1971 2),3), we wonder whether we are reinventing the wheel; since logically components having the largest number of lines are also the most complex, and the ones with more frequent changes or past faults. Moreover, previous research works have reported inter-correlation among some of these predictors 3).

If our supposition is right, used predictors would show inter-correlation among them, and the gain of having more than one of these metrics in the same model (multivariate) would not be significant. We performed a rapid literature review trying to answer the following questions: If inter-correlation is observed, how much more accurate are multivariate models than univariate models? and how much does their usage worth? If a multivariate model yields a number of accurate predictions roughly identical to the number yielded by a cheaper model such a LOC based model, then no significant gain can be considered. The remainder of this paper is organized as follows: First, we present some background, then we explain our methodology and findings, followed by a proposed course of action and conclusions.

## 2. Background

Multicollinearity is presented when a number of predictor variables are highly intercorrelated. Naive Bayes and Logistic regression are reported to be the techniques that are performing relatively well for fault prediction 1). Both assume the predictors be independent of each other 4)5).

## 3. Methodology and Findings

We analyzed some of the results provided by Hall et al. in 1), where 208 research works published during 2000-2010 were reviewed, and from which only 36 studies reported sufficient contextual and methodological information according to the reviewers criteria. We found that:

- 139 of 172 different datasets used accross the reviewed works were from open source projects (mainly Eclipse).
- LOC performed overall well and, in some cases, better than other sets of metrics such as OO metrics.
- The topic of multicollinearity among predictors variables is not discussed.

Next, we examined 13 studies of their 36 selected papers[⋆1], and we found that:

- In general, they make reference to the per-

†1 Graduate School of Information Science
Nara Insititute of Science and Technology

⋆1 We refer to these using the same reference numbers as in Hall et al. 1), where their full bibliography can be found.

formance of prediction models based on sets of metrics (e.g. size, OO and process). No details are provided about single predictors or correlation within the sets.

- Only one study, [[51]], shows a correlation matrix among the predictor variables (the Chidamber and Kemerer set), and studies the prediction accuracy of univariate and multivariate models. Their results suggest that a univariate model using CBO as a predictor (Coupling Between Objects) performs the best.
- Five studies, [[9]], [[18]], [[31]], [[32]] and [[37]], state a correlation problem among their predictors, but they do not give major details. They do state the method applied to address this problem; four used principal component analysis and one correlation-based-future selection.
- Two studies, [[32]] and [[56]], suggest LOC as a useful predictor and with stable performance. Another, [[11]], reported LOC to have poor predictive power, although in their previous work by the same authors LOC-only model was suggested as viable alternative to more complex models.
- Studies [[8]], [[10]], [[18]], [[29]] and [[69]] did not provide any or significant information about correlation among predictors.
- Authors of studies [[9]] and [[8]] propose a measure of cost-effectiveness of prediction models, which is based under the assumption that the cost of testing a class is roughly proportional to its size. Therefore, if the only thing a prediction model does is to model the fact that the number of faults of a class is proportional to its size, they say, there would be likely no much gain from such a model. Their conclusions are over sets of metrics; no details are given about single predictors.

Due to the lack of information provided by the authors, we cannot answer certainly our posed questions, but we cannot reject our supposition either, since single metrics such as LOC or CBO have been reported to be more accurate than sets of metrics.

Recently, other metrics have been suggested as good predictors, such as process and socio-technicals metrics (based on developer's contribution and components dependency), both mined from logs of code repositories; yet, neither details are given about multicollinearity, nor univariate analysis is provided.

## 4. For Discussion: Course of action

Study of other metrics. Since most of the research on this field has been done using open source projects, we think that product metrics have been explored exhaustively enough as predictors of faulty code. Trying to find a prediction model that explains fault-proneness of code using *only* these kind of metrics *leaves out* the possibility of explaining fault-proneness of code due to other more fundamental factors underlying the generation of faulty code such as requirements misinterpretations, developer and designers' experience, and management skills. But, which data, of this kind, is publicly available for research?

Cost-effectiveness measures. Since testing activities are time-consuming, there is a need to evaluate models not only by considering their prediction accuracy, but also by assessing the potential cost-effectiveness of applying such models, as Arisholm et al. suggested in [[9]]. And, we would like to add that these such include not only multivariate, but also univariate analysis of the different proposed metrics.

## 5. Conclusion

We found that although multicollinearity among predictors of faulty code is reported, little is reported about the gain of using these in multivariate models as opposed to univariate models. We think that researchers have exhausted the exploration of product metrics from open source software and other factors which may be related to faulty code are difficult to study due to the lack of publicly available data.

## References


1) Hall, T., Beecham, S., Bowes, D., Gray, D. and Counsell, S.: A Systematic Literature Review on Fault Prediction Performance in Softw. Eng., *Softw. Eng., IEEE Trans. on*, Vol. 38, No.6, pp.1276–1304 (2012).
2) Akiyama, F.: An Example of Softw. System Debugging, *IFIP Congress*, pp.353–359 (1971).
3) Fenton, N.E. and Neil, M.: A Critique of Software Defect Prediction Models, *IEEE Trans. Softw. Eng.*, Vol.25, No.5, pp.675–689 (1999).
4) Sharma, S.: *Applied Multivariate Techniques*, Addison-Wiley & Sons, Inc., USA (1996).
5) Lewis, D. D.: Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval, *Proc. of the 10th Eur. Conf. on Machine Learning*, ECML '98, London, UK, UK, Springer-Verlag, pp.4–15 (1998).