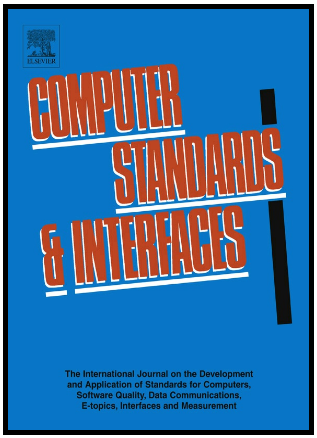
ELSEVIER
COMPUTER
STANDARDS
& INTERFACES
The International Journal on the Development and Application of Standards for Computers, Software Quality, Data Communications, E-topics, Interfaces and Measurement

www.elsevier.com

An Empirical Analysis of the Effectiveness of Software Metrics and Fault Prediction Model for Identifying Faulty Classes

Lov Kumar, Sanjay Misra, Santanu Ku. Rath

PII: S0920-5489(16)30088-5
DOI: http://dx.doi.org/10.1016/j.csi.2017.02.003
Reference: CSI3201

To appear in: *Computer Standards & Interfaces*

Received date: 20 September 2016
Revised date: 27 December 2016
Accepted date: 10 February 2017

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2017) 1–40

Procedia Computer Science

An Empirical Analysis of the Effectiveness of Software Metrics and Fault Prediction Model for Identifying Faulty Classes

Lov Kumar[a], Sanjay Misra[b], Santanu Ku. Rath[c]

[a]*Dept. CSE, National Institute of Technology , Rourkela, India*
lovkumar505@gmail.com
[b]*Atilim University, Ankara, Turkey*
ssopam@gmail.com
[c]*Dept. CSE, National Institute of Technology , Rourkela, India*
skrath@nitrkl.ac.in

Abstract

Software fault prediction models are used to predict faulty modules at the very early stage of software development life cycle. Predicting fault proneness using source code metrics is an area that has attracted several researchers' attention. The performance of a model to assess fault proneness depends on the source code metrics which are considered as the input for the model. In this work, we have proposed a framework to validate the source code metrics and identify a suitable set of source code metrics with the aim to reduce irrelevant features and improve the performance of the fault prediction model. Initially, we applied a t-test analysis and univariate logistic regression analysis to each source code metric to evaluate their potential for predicting fault proneness. Next, we performed a correlation analysis and multivariate linear regression stepwise forward selection to find the right set of source code metrics for fault prediction. The obtained set of source code metrics are considered as the input to develop a fault prediction model using a neural network with five different training algorithms and three different ensemble methods. The effectiveness of the developed fault prediction models are evaluated using a proposed cost evaluation framework. We performed experiments on fifty six Open Source Java projects. The experimental results reveal that the model developed by considering the selected set of source code metrics using the suggested source code metrics validation framework as the input achieves better results compared to all other metrics. The experimental results also demonstrate that the fault prediction model is best suitable for projects with faulty classes less than the threshold value depending on fault identification efficiency (low- 48.89%, median- 39.26%, and high- 27.86%).

Keywords: Feature selection techniques, Artificial neural network, Ensemble method, Source code metrics, Cost analysis framework.

1. INTRODUCTION

Most software companies intend to apply object-oriented (OO) technology for building modern software systems due to its effective design features such as enhancing code re-use and reducing fragility, which enable faster product development. OO Paradigm also assists in optimal characterization (e.g., abstraction, encapsulation, inheritance, and polymorphism) of the software system compared to other paradigms. Along with these design strategies, one of the major objectives of a software companies is to estimate and improve the quality of software; in general. They further adopt certain processes, such as deployment of bug-tracking systems, code reviewing, proper testing to ensure software quality, and reduce faults in a system. However, ensuring a system to be entirely fault-free is not practically