ABSTRACT

Bug severity is the degree of impact that a defect has on the development or operation of a component or system, and can be classified into different levels based on their impact on the system. Identification of severity level can be useful for bug triager in allocating the bug to the concerned bug fixer. Various researchers have attempted text mining techniques in predicting the severity of bugs, detection of duplicate bug reports and assignment of bugs to suitable fixer for its fix. In this paper, an attempt has been made to compare the performance of different machine learning techniques namely Support vector machine (SVM), probability based Naïve Bayes (NB), Decision Tree based J48 (A Java implementation of C4.5), rule based Repeated Incremental Pruning to Produce Error Reduction (RIPPER) and Random Forests (RF) learners in predicting the severity level (1 to 5) of a reported bug by analyzing the summary or short description of the bug reports. The bug report data has been taken from NASA’s PITS (Projects and Issue Tracking System) datasets as closed source and components of Eclipse, Mozilla & GNOME datasets as open source projects. The analysis has been carried out in RapidMiner and STATISTICA data mining tools. The authors measured the performance of different machine learning techniques by considering (i) the value of accuracy and F-Measure for all severity level and (ii) number of best cases at different threshold level of accuracy and F-Measure.

Keywords: 10-fold Cross Validation, Bug Repositories, Bug Severity, Multiclass Classification, Supervised Classification, Text Mining

DOI: 10.4018/jossp.2013040103
INTRODUCTION

With the increasing use of software in every sphere of life, it is often found that software does not function properly or some of the functionalities need minor change for improvement. It is becoming very important to record these problems or changes using suitable bug reporting system. A bug or fault is a program defect that is encountered while operating the product either under test or in use. In order to provide the details of bug to the development team, different bug tracking systems are being used by the industry. The bug tracking systems are helpful in bug reporting as well as tracking the progress of bug fixes. Various bug tracking systems have been proposed in the available literature that includes Bugzilla (http://www.bugzilla.org/), Jira (http://www.atlassian.com/software/jira/), Mantis (http://www.mantisbt.org/), etc. Recently, a bug tracking and reliability assessment tool has been proposed (Singh & Chaturvedi, 2011), which helps in tracking the progress of fix as well as reporting the bug. Various parameters are filled up by the tester or user who reports the bug. During bug reporting, different bug attributes namely bug title, short description or summary, detailed description, priority, severity are filled up. The value of severity and priority may not be accurate as far as the submitted report is concerned because users may not have complete information about the modules/ components in which the bug has occurred. Priority of a reported bug represents the urgency of its fix; accordingly a number or level associated with priority has to be assigned. There are five different numbers or levels defined for priority. In addition to the priority, another important parameter that affects the priority of the reported bug is severity. Severity is defined as the impact of bug in working functionality of a component or the system. The impact of the bug varies from user to user. Generally, people do require that their bug must be taken on a priority basis irrespective of the impact on the user or the developer or the system itself. Software projects have clear guidelines on how to assign a severity level to a bug. But, due to non-awareness, people often commit mistakes in assigning the severity level during bug reporting. The severity level can be categorized broadly in five to seven categories. Bug repository of closed source projects of NASA’s PITS (Projects and Issue Tracking System) dataset has five severity levels vary from the fullest to the dullest or from 1 to 5. Bug repositories of open source projects have defined seven severity levels, and these severity levels vary from the blocker, critical, major, enhancement, minor, normal, to trivial or from 1 to 7. Level 1 of severity represents fatal errors and crashes whereas level 5 or 7 of severity mostly represents cosmetic changes such as formatting, alignment, comments and display messages. Other severity levels are assigned for bugs due to addition of new features and enhancement of the existing feature. The default value “Normal” is normally assigned by most of the reporters because they do not analyze the bug seriously. Severity is a critical factor in deciding the priority of a reported bug. The numbers of reported bugs are usually quite high, hence, it is becoming necessary to have a tool/technique which can determine or verify the severity of a reported bug. It is the need of the hour to automate the process of determining the bug severity for the reported bug. Our work is based on the following research questions:

Research Question 1: Are the Machine Learning Techniques applicable in predicting the severity level of a reported bug?

Research Question 2: What is the order of applicability of different machine learning techniques on the basis of different performance measures?

Research Question 3: Is there any effect of number of terms on the performance of different Machine Learning Techniques?

To answer the above research questions, a study has been conducted by applying the Naive Bayes, Decision Tree, RIPPER, Random Forest and Support Vector Machine in the bug repositories of open as well as closed source projects. The performance measures namely accuracy, precision, recall and F-Measure have
Making Knowledge Management Systems Open: 
A Case Study of the Role of Open Source Software
Tom Butler, Joseph Feller, Andrew Pope and Ciaran Murphy (2007). Open Source for Knowledge and Learning Management: Strategies Beyond Tools (pp. 150-174).
www.igi-global.com/chapter/making-knowledge-management-systems-open/27811?camid=4v1a

Teaching Software Engineering with Free/Libre Open Source Projects
www.igi-global.com/article/teaching-software-engineering-free-libre/2772?camid=4v1a