A New Data Mining-Based Framework to Test Case Prioritization Using Software Defect Prediction

Emad Alsukhni, Department of Computer Information Systems, Yarmouk University, Irbid, Jordan
Ahmad A. Saifan, Department of Computer Information Systems, Yarmouk University, Irbid, Jordan
Hanadi Alawneh, Department of Computer Information Systems, Yarmouk University, Irbid, Jordan

ABSTRACT

Test cases do not have the same importance when used to detect faults in software; therefore, it is more efficient to test the system with the test cases that have the ability to detect the faults. This research proposes a new framework that combines data mining techniques to prioritize the test cases. It enhances fault prediction and detection using two different techniques: 1) the data mining regression classifier that depends on software metrics to predict defective modules, and 2) the k-means clustering technique that is used to select and prioritize test cases to identify the fault early. Our approach of test case prioritization yields good results in comparison with other studies. The authors used the Average Percentage of Faults Detection (APFD) metric to evaluate the proposed framework, which results in 19.9% for all system modules and 25.7% for defective ones. Our results give us an indication that it is effective to start the testing process with the most defective modules instead of testing all modules arbitrary arbitrarily.

KEYWORDS

Data Mining, Software Defect Prediction, Software Testing, Test Case Prioritization

INTRODUCTION

Testing is the most critical phase in any software development life cycle (SDLC). Different objectives are achieved through testing process including, but not limited to, assuring the functionality of software for customers; showing the existence errors; and ensuring the quality of the product. Regression testing is a testing approach that is performed after modifying the software during the maintenance phase. It can detect new errors after making some changes in the features of the software. However, re-executing all the regression test cases is expensive and time-consuming. Hence, different methods are proposed to efficiently and effectively execute regression testing, namely, test case minimization, selection, and prioritization.

Prioritization process uses different techniques to arrange test cases to help early fault detection and increase the rate of code coverage (Maheswari & JeyaMala, 2013; Pathania & Kaur, 2015; Pradeepa, & VimalaDevi, 2014). Moreover, data mining techniques can utilize different classification
and clustering approaches to predict the most defective modules using different software metrics (Yousef, 2014).

Testing is a critical phase in the software development process since it consumes 50% of the development cost and resources in terms of effort and time (Sommerville, 2010). Testers have to execute several different test cases in order to detect faults, which demands additional cost and time. On the other hand, not all test cases have the same ability to detect faults in software, which means that some test cases must be executed before others (Mary & Amoorthi, 2011). To handle this limitation, we introduced a framework to prioritize the most important test cases that can detect faults in a shorter time and incurring less cost than randomly running all test cases.

The main objective of this research is to combine classification and clustering techniques to prioritize the test cases based on the software defect prediction model (SDP) with a view to reducing time, cost and effort in the testing process. This can be done by starting the testing process from the most defective modules rather than testing modules arbitrarily. The prediction of defective modules is based on software quality product metrics. We investigated a hybrid data mining approach, which combines classification and clustering algorithms to prioritize test cases based on SDP. The paper explores an answer to the research question “Can the SDP model improve test prioritization technique?”

The rest of the paper is organized as the following: The related work is discussed in the following section. The proposed framework is presented in section 3. Experiments and analysis are discussed in section 4. Section 5 discusses the threats to validity. Finally, the paper is concluded in section 6.

LITERATURE REVIEW

SDLC is a process of designing, implementing and modifying software. Software development is a human process; it is quite obvious that it is difficult to develop completely fault-free software. Moreover, software complexity and many other constraints may introduce or raise more errors. The SDP is used at the earliest stage of the development in order to reduce development and maintenance costs, effort, customer dissatisfaction, and project failure. In this section, we review some of the previous works on SDP and test case prioritization techniques. For this purpose, the concepts of data mining and SDP are combined to introduce the SDP data mining technique. Data mining is an approach for finding useful knowledge and common patterns in different data sources involving data transactions, databases and data warehouses. The main goal of the data mining process is to extract useful and meaningful information from large amounts of data. Moreover, software metrics as an indicator for defective modules; regression testing techniques and evaluation metrics; test case prioritization using data mining techniques are also introduced.

SDP Using Data Mining

Software Defect Prediction (SDP) is the cognitive operation of locating defective modules in software (Baindara & Yadav, 2014). Applying SDP at the earliest stage of the software development process can save cost, effort, and has been shown to produce reliable software. Software organizations can focus on fault prediction models to manage their testing resources effectively through focusing on fault-prone software modules extracted from software repositories to improve software quality.

Generally, software defect prediction models find the number of defects or errors in system modules or classes, or categorize system classes into defective and non-defective (Gayathri & Sudha, 2014; Karpagavadiyvu et al., 2012).

Software defect prediction models find the number of defects or errors in system modules or classes, or categorize system classes into defective and non-defective classes according to different software quality metrics such as: line of code, Cyclomatic complexity, branch count, etc. These measurements can be directly extracted from the code (Megal & Jacob, 2015).

Clustering and classification techniques have become the most common methods for fault prediction and detection. Yousef (2014) investigates 22 attributes from five software projects which
Virtualized Open Source Networking Lab
Lee Chao (2013). Open-Source Technologies for Maximizing the Creation, Deployment, and Use of Digital Resources and Information (pp. 85-100).
www.igi-global.com/chapter/virtualized-open-source-networking-lab/70120?camid=4v1a

VuFind: Solr Power in the Library
www.igi-global.com/chapter/vufind/120981?camid=4v1a