An Efficient Software Defect Analysis Using Correlation-Based Oversampling

K. Nitalaksheswara Rao · Ch. Satyananda Reddy

Received: 2 May 2017 / Accepted: 30 December 2017 / Published online: 16 January 2018
© King Fahd University of Petroleum & Minerals 2018

Abstract
In software industry, probability of risks in large project is increasing due to the presence of defective modules which leads to failure in software execution. Quality assurance is one of the important aspects of software development. The extracted information from software repositories about defective modules can help project managers a lot for completion of projects on time with reliable quality. There are many existing knowledge extracting techniques for software defect analysis and prediction, but they have considered this problem in a general framework. Class imbalance is one of the major problems for constructing efficient decision trees for extracting knowledge from software defect datasets. To overcome this problem, a novel approach called as ICOS (Improved Correlation over Sampling) is proposed for handling class imbalance software defect datasets. This proposed approach uses oversampling strategy to generate new instances using synthetic and hybrid category approaches. The experimental results confirm that the proposed approach can efficiently identify the modules which are error-prone using simple rules.

Keywords Software defects analysis · Classification · Decision tree · Class imbalance learning · Oversampling

1 Introduction

The software development process is a complex and tedious process. In the life cycle of software engineering, the most costly and difficult phase is the testing phase. The exhaustive testing of the complete software is very large task for the quality assurance team. The crucial factors such as time, cost and need of more manpower will not permit the exhaustive testing of the complete software. Hence, expert advice or oracle services are needed for the effective testing phase of the software development. The knowledge discovery process can work as a virtue for solving the above-discussed problem.

In the process of knowledge discovery from datasets, there are two broad approaches such as supervised and unsupervised learning. In supervised learning the task is to predict the unforeseen case by using the model built from the existing cases. In unsupervised learning, the set of cases are categorized into different groups by using the intrinsic characteristics and properties of the available cases. The scenario of software defect analysis cannot be classified as an ordinary case, since it comes under a special class known as class imbalance learning. This is one of the historical limitations or mistakes done by the research community in the software defect analysis [1].

In the real scenario of software dataset, the researcher can observe that the modules or instances of the software are defective in a small ratio. The defective or non-defective classes of any real-time scenario software dataset are of class imbalance nature, i.e., in one class there are more percentage of instances than other class. A complete study regarding imbalance learning can be obtained from [2]. Figure 1 presents the visualization of four class imbalance software defect datasets: ar1, kc1-class-level-defective, kc1-class-level-top5per and reuse dataset. In this paper, a novel approach for efficient knowledge discovery for software defect datasets is proposed. The proposed approach improves the defect prediction accuracy, thereby reducing the manpower for quality assurance stage of the software development process. Along with that, we have investigated...
three crucial research questions with detailed analysis for class imbalance software defect prediction.

The rest of this paper is organized as follows: Sect. 2 presents the related work for software defect analysis. Section 3 describes the problem statement. Section 4 presents the proposed methodologies and algorithm, respectively. In Sect. 5, the details of experimental setup are presented. Simulation results are discussions and analyzed in Sect. 6 and conclusion is presented in final section.

2 Related Work

There is a rich literature related to the topic of software defect prediction. In this section, we presented the identified most closely related work.

The traditional software defective analysis techniques are not capable of efficiently handling imbalance datasets. In recent years, researchers in different community made efficient proposals on imbalance datasets. Liu et al. [3] proposed an efficient approach for handling class imbalance data in the domain of customer behavior prediction. Perez-Sanchez et al. [4] have presented a unique approach for handling class imbalance problem using target concept for one class classification. Liu and Forss [5] have presented an improved approach to handle class imbalance issue in the scenario of text classification for web content filtering and safety. Barnab'e-Lortie et al. [6] have proposed a novel approach for active learning of class imbalance data for one class classification using labeling the instances in priority. Lakshmi and Prasad [7] have presented different approaches using SMOTE to handle the problem of class imbalance.

Alvarez et al. [8] have proposed methods for supervised and unsupervised methods for making management rules easier for development process of software. Deep et al. [9] have proposed methods for efficient software engineering development with a focus on how knowledge extracting techniques help in achieving the software engineering goals and benefit the software engineering tasks. Husain et al. [10] have studied some techniques to be applied in solving software engineering problems. These techniques are applied to detect problems such as bugs, to aid in pattern discovery and to help developers deal with the complexity of existing software, in order to create more failure-free software. Azeem et al. [11] have provided description about software defect prediction and describe the key areas of software defect prediction practice, and highlight some key open issues for the future.

Yasmeen et al. [12] have presented techniques for practitioners and researchers which can be used for exploring valuable data in order to better manage their projects and do produce higher-quality software systems that are delivered on time and within budget. Jai kaur and Pallavi [13] have discussed techniques for software defect prediction which helps the developers to detect software defects and correct them. Padmabhushana et al. [14] have introduced an extensive research framework capable of preprocessing web log bug data completely and efficiently. The framework reduces the error rate and improves significant learning performance of the algorithm. The framework helps to investigate the software bug behavior efficiently using Naïve Bayesian classifier as the base algorithm. Das et al. [15] have presented an improvised core tool used to facilitate software engineering research on large and diverse dataset.

Kumar et al. [16] have developed a framework for fault prediction using the statistical methods for embedded software systems. Ceren et al. [17] have developed an automatic refining system for activity recording for possible execution paths in test mode. The refined model is updated automatically when new test cases generates for more effective testing. Chen et al. [18] have proposed a model for ensemble learning for improving defect prediction using neighbor cleaning and random under sampling techniques.

The C4.5 [19] decision tree is capable of accurately classifying the majority (more instances) class which is usually the less important class, while the accuracy of minority (less instances) class drops drastically when compared to majority class. All existing techniques do not focus on generation of best rules to achieve better accuracy. The work published by Wang et al. [20] have conducted benchmark experimental setup for answering the specific research questions regarding the issues related to class imbalance and software defect prediction. According to the authors’ opinion, still improvement and further investigation are required for more detailed analysis. Hence, novel approaches are needed to make efficient knowledge discovery from the software defect datasets. The above literature studied provided the confidence and motivation to propose novel algorithm in the field of software engineering using data mining techniques.

3 Problem Statement

In the research community, there are many data sources which could be mined to obtain the required knowledge for software defect analysis. The knowledge discovered from these data sources can be used for performing the following tasks:

1. Identifying the software modules which are error-prone.
2. Estimating the software defects in hand.
3. Finding the causes and solution for the software defect modules.
4. Assigning defective modules to the suitable developers.

In the process of above findings all the existing classifiers consider class imbalance as the general problem but due to the