3rd.International Conference on
Theoretical and Applied Computer Science and Engineering
(ICTACSE, 2018)
Ankara, Turkey, November 23-24,2018

A Stepwise Approach for Bug Fix Time Prediction

Musa Usumus
Faculty of Computer and Informatics Engineering,
Istanbul Technical University
Istanbul, Turkey
usumus@itu.edu.tr

Ayse Tosun
Faculty of Computer and Informatics Engineering,
Istanbul Technical University
Istanbul, Turkey
tosunay@itu.edu.tr

***Abstract*— Software maintenance is a challenging process, and requires efforts of both developers and managers. One issue for related to this process is software bugs that break the normal flow of the software. There are many studies in the literature about software bugs, e.g., bug triaging, bug prediction, bug fix time prediction, duplicate bug detection. In this study, we focus on predicting the fix time for a newly reported bug based on historical bug reports stored in issue repositories. Bug fix time has been predicted either using previous bug's attributes or bug's textual descriptions in the prior studies. In this study, we combine both bug's attributes and textual description information in a stepwise approach: First, we use the textual similarity between bugs' descriptions to find the closest bug set. Second, we use these closest bugs and use their bug attributes to build machine learning based prediction models. The stepwise model is compared with the other two models which use each information separately. We also define the fix time different than the traditional approaches. For an incoming bug report, we can classify it as slow, very slow, fast or very fast fix, by applying quartile based discretization on fix time. All models are built and validated on an open source software project, Mozilla, and statistical tests are used to evaluate the performance of the stepwise model over the others. The proposed stepwise model slightly improves the bug fix time prediction on Mozilla dataset even though its accuracy needs to be further improved. According to the statistical tests, using textual information with the bug attributes in a stepwise approach does not have a noticeably greater impact on accuracy compared to the other models.**

Keywords—software bugs, bug resolution/fix time prediction; software maintenance; stepwise model

I. INTRODUCTION

With the increasing use of software systems, maintenance has become a really challenging process. In large-scale software organizations, teams spend a lot of time and effort prior to deployment for finding and fixing residual bugs that may break or block the normal flow of the software execution. Yet, there may be many failures reported on a daily basis to the bug tracking systems of such organizations. It is shown that buggy software reduces customer value, and it is important to deliver software versions that have fewer bugs for customer sustainability [3]. There have been many related work in the literature about software bugs, e.g., bug prediction, bug triaging, predicting the fix time of the bug, checking duplicate bugs, predicting the number of bugs will be opened in the next releases. For example, Tamrawi et al. [20] provide fuzzy set-based approach for automatic bug triaging. Nguyen et al. [16] use information retrieval and topic modelling to detect duplicate bug reports. Zhang et al. [24] analyze the delays occurred by developers in bug fixing process. Bhattacharya and Neamtiu [4] shows that the reputation of the reporter does not have an effect on the fixing time. Canfora et al. [10] analyze source code constructs with the bug survival time, and results indicate some code blocks are correlated with the bug lifetime, i.e., changing exception handling constructs with the control flow statements correlate with high survival time. *In this study, we focus on predicting the time to resolve a bug.* The bug fix time is generally expected to be associated with the bug attributes. For example, a bug having critical severity is expected to be resolved quicker than those with other severities [24]. Some studies predict bug fix time as a categorical value (e.g. [25]), while others as continuous (e.g. [2]). Simple to more complex machine learning algorithms are also utilized to predict bug fix time using bug attributes, e.g. kNN-based model in [25], Hidden Markov Model in [14], or other techniques using textual bug descriptions, e.g. [23].

In this paper, we combine approaches proposed in prior studies, and propose a stepwise approach on predicting bug fix time. We initially classify bugs as *very slow, slow, fast, very fast* in an open source project. During the prediction, we first use textual similarity techniques based on bugs' textual descriptions to find the closest group of bugs to our test data. Using these filtered bug group, we perform a second step based on bugs' attributes other than textual information and predict bug time using Random Forest algorithm. We then evaluate the performance of the proposed approach with the models using bug attributes or textual information only.

This paper is organized as follows. Section II summarizes related work on bug fix time prediction. Section III presents the dataset and associated information. Section V explains model construction. Section VI reports the experimental results. We interpret our findings in Section VII and discuss the threats to results' validity in Section VIII. Section IX concludes the paper.

II. RELATED WORK

Bug fix time can be modelled as a continuous [2, 25] or categorical value [16, 27]. Also, studies investigating bug fix time prediction can be classified in terms of the features used: textual attributes, bug attributes or both.

Previous studies report that bug attributes can be used to predict fix time as a categorical value (e.g. *slow, fast*) or as a continuous value (e.g. 85 hours). Anbalagan and Vouk [2] are investigated the relation between the number of people