International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 1 | ISSN : 2456-3307

Survey of Various Data Reduction Methods for Effective Bug Report Analysis

Kapil Sahu[1], Dr. Umesh Kumar Lilhore[2], Prof. Nitin Agarwal[3]
[1]M. Tech Scholar, Department of CSE, NIIST Bhopal, Madhya Pradesh, India
[2]Head PG, Department of CSE, NIIST Bhopal, Madhya Pradesh, India
[3]Assistant Professor, Department of CSE, NIIST Bhopal, Madhya Pradesh, India

ABSTRACT

In software development process testing process ensures quality management of the product by ensuring bugs free product. During software development and testing process, lots of bugs are logged, fixed and reopened. Bugs management is always expensive and time consuming for software companies. Bug reports are essential software artifacts that describe software bugs, especially in open-source software. Lately, due to the availability of a large number of bug reports, a considerable amount of research has been carried out on bug-report analysis, such as automatically checking duplication of bug reports and localizing bugs based on bug reports. In particular, this paper first presents some background for bug reports and gives a small empirical study on the bug reports to motivate the necessity for work on bug-report analysis. Then this paper summaries the existing work on bug-report analysis and points out some possible problems in working with bug-report analysis

Keywords: Data Mining, Bug Data Reduction, Bug Report Analysis, Data Management in Bug Repositories.

I. INTRODUCTION

Data mining has been introduced to the technically developing environment as a promising means to handle the software data. By using the data mining techniques, mining software repositories can uncover interesting and hidden information of the software repositories and can also solve the real world software problems. Many software companies spend almost half of their project money in fixing the bugs [7]. Large software projects have bug repository that holds all the information related to bugs and is well maintained for further processing. In bug repository, each software bug has a bug report and is also known by bug data.

The bug report consists of textual information of the bug and the updates on the basis of the status of bug fixing, which is available in historical bug dataset [1]. Traditional software analysis is not fully suitable for the large-scale and complex data in software repositories [4].

In this paper, we are presenting a survey of various data reduction methods for effective bug report analysis. This paper is organized into various chapters such as introduction, bug reports, and data mining, related work, challenges in the existing system and finally described conclusion and future work.

II. BUG TRIAGE & DATA MINING

Bug triage is a process where tracker issues are screened and prioritized. In software development process bug triage plays a vital role. Manual bug triage by a human trainer is time-consuming and error-prone since the number of daily bugs is large and lack of knowledge in developers about all bugs. Because of all these things, bug triage results in expensive time loss, high cost, and low accuracy.

In some former methods, if a bug report is formed or a bug occurred, then a human triager assigns this bug to a developer, who tries to fix this bug. This developer is recorded in an item assigned to in historical bug dataset. If the previously assigned developer was unable to fix this bug then the former will change to a new one.

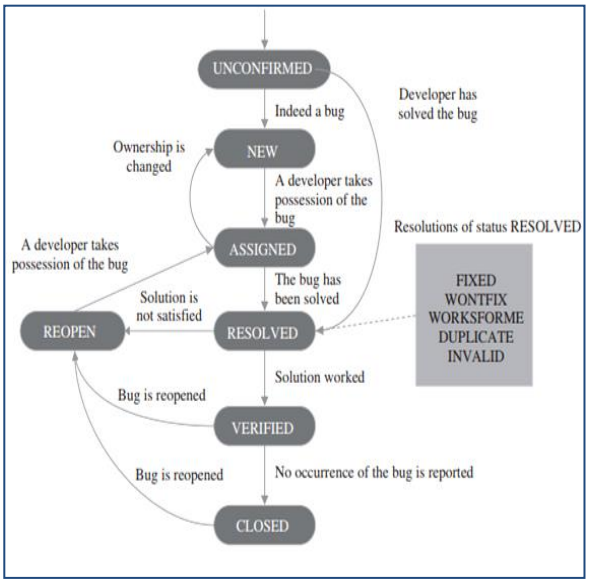

Figure 1. Bug Life Cycles

The method of assigning a proper developer for fixing the bug is known as bug triaging (figure 2.1). When a bug is first reported, the bug report is marked as UNCONFIRMED. When a triager has verified that the bug is not duplicate and indeed a new bug, the status is set to NEW.

Then the triager assigns the bug report to one proper developer, and the status is changed to ASSIGNED. Then the assigned developer reproduces the bug, localizes it and tries to fix it. When the bug has been solved, the bug report is marked as RESOLVED. After that, if a tester is not satisfied with the solution, the bug should be reopened with the status set to REOPEN; if a tester has verified that the solution worked, the status is changed to VERIFIED. The final status of a bug report is CLOSED, which is set when no occurrence of the bug is reported.

Data mining has been introduced to the technically developing environment as a promising means to handle the software data. By using the data mining techniques, mining software repositories can uncover interesting and hidden information of the software repositories and can also solve the real world software problems such as bug triage. In particular, these approaches usually train a classifier with previously assigned bug reports and then use the classifier to classify and assign new bug reports.

2.1 Clustering: Clustering is a form of unsupervised learning in which no class labels are provided. It is often the first data mining task applied to a given collection of data. In this, data records need to be grouped based on how similar they are to other records. It is a task of organizing data into groups such that the data objects that are similar to each other are put into the same cluster. The groups are not predefined. It is a process of partitioning a data in a set of meaningful sub-classes called clusters. Clusters are subsets of objects that are similar. Clustering helps users to understand the natural grouping or structure in a data set. Its schemes are evaluated based on the similarity of objects within each cluster.

2.2 Classification: Classification is a process of finding a set of models that describe and distinguish data classes or concepts. It is the organization of data in given classes known as supervised learning, where the class labels of some training samples are given. These samples are used to supervise the learning of a classification model. Classification approaches normally use a training set where all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model. The model is used to classify new objects.

2.3 Association: The Association mining task consists of identifying the frequent itemsets and then forming conditional implication rules among them. It is the task of finding correlations between items in data sets. Association Rule algorithms need to be able to generate rules with confidence values less than one. Association rule mining is undirected or

unsupervised data mining over variable-length data and it produces clear understandable results. The task of association rules mining consists of two steps. The first involves finding the set of all frequent itemsets. The second step involves testing and generating all high confidence rules among item sets.

III. EXISTING WORK

In this survey work, we have studies various research work suggested by different authors for bug report analysis using different data reduction methods.

Sangameshwar Patil et al. presented a concept based classification of software defect reports. Author [1] proposed the use Explicit Semantic Analysis (ESA) to carry out the concept-based classification of software defect reports. Finally compute the "semantic similarity" between the defect type labels and the defect report in a concept space spanned by Wikipedia articles and then, assign the defect type which has the highest similarity with the defect report. This approach helps us to circumvent the problem of dependence on labeled training data.

Attika Ahmed et al. presented An Improved Self-Organizing Map for Bugs Data Clustering. In this work author [2] attempts to provide a comparative analysis of both the clustering algorithms and for attaining the results, a series of experiment has been conducted using Mozilla bugs data set. In software projects, there is a data repository which contains the bug reports. These bugs are required to carefully analyze and resolve the problem. Handling these bugs humanly is an extremely time-consuming process, and it can result in the delaying in addressing some important bugs resolutions. To overcome this problem, researchers have introduced many techniques. One of the commonly used algorithms is K-means, which is considered as the simplest supervised learning algorithm for clustering, yet it tends to produce smaller number of clusters, while considering the unsupervised learning algorithms, Self-Organizing Map (SOM) considers the equally compatible algorithm for clustering, as both the algorithms are closely related but differently used in data mining.

Dhyan Chandra et al. worked on "Software Bug Detection using Data Mining". The common software problems appear in a wide variety of applications and environments. Some software related problems arise in software project development i.e. software related problems are known as software defect in which Software bug is a major problem arises in the coding implementation.There is no satisfying result found by project development team. The software bug problems mentation in problem report and software engineer does not easily detect this software defect but by the help of data mining classification software engineers easily can classify software bug. This work [3] classified and detect software bug by J48, ID3, and Naive Bayes data mining algorithms.

Suman, Seema et al. worked on Classification of Bug Reports Using Text Mining. In this author [4] proposed a new approach, in which they calculate the average length of the terms in the synonym list. After calculating the average, they have considered only those terms which are having a length greater than or equal to the calculated average length. The terms having a length less than the threshold value are not considered during classification. This way they have reduced the number of terms which are to be matched with the synonym. So by ignoring the rest of the terms, we have saved a significant amount of time. The trigger is the person who manually labels the bug.

R. Pon Periasamy et al. worked on Data Mining Techniques in Software Defect Prediction. The main objective of paper [5] is to help developers identify defects based on existing software metrics using data mining techniques and thereby improve the software quality. In this paper, the author discussed data mining techniques that are association mining, classification, and clustering for software defect

prediction. This helps the developers to detect software defects and correct them.

Kirti Shamrao Tandale1 et al. worked on A Survey on Effective Bug Triage with Data Reduction. In this work author [6] primarily focus the bug reduction system in this project with an assumption that the communication channel between the developer and the bug reduction is maintained. The author also prevents a redundant bug in the repository. They have introduced a novel alternative that provides a significantly-improved bug report. Users dislike the redundancy of the same bug frequently in the bug data and assign an appropriate developer to resolve bug issues. The second approach allows the associated developer to resolve them according to bug classification. This is a tedious assumption since private data can be exposed by either software bugs or configuration errors at the trusted servers or by malicious administrators. Finally, it is relying on heavy-weight mechanisms to obtain provable redundant bug report.

IV. CHALLENGES

The information stored in bug reports has two main challenges.

1. **Firstly the large-scale data** - Due to a large number of daily reported bugs, the number of bug reports is scaling up in the repository.
2. **Secondly, low quality of data-** Noisy and redundant bugs are degrading the quality of bug reports

V. CONCLUSION & FUTURE WORK

One of the expensive steps in software maintenance is Bug Triaging, mainly when it comes to the matter of labor and time cost. The recent technique aims to form reduced and high-quality bug data in software development and thereby maintenance. The data processing techniques like instance selection and feature selection are used for data reduction. In this paper, we have presented a survey on effective bug report analysis method by using data reduction and data mining methods.

In future work, we will propose and developed an efficient data reduction method for bug triage to reduce the software development and total maintenance cost.

VI. REFERENCES

[1]. Sangameshwar Patil, "Concept-based Classification of Software Defect Reports", IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), May 2017, PP 182-187.

[2]. Attika Ahmed, Rozaida Ghazali,"An Improved Self-Organizing Map for Bugs Data Clustering", IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS) Malaysia, October 2016, pp 135-142.

[3]. Dhyan Chandra Yadav, Saurabh Pal,"Software Bug Detection using Data Mining", International Journal of Computer Applications (0975 – 8887) Volume 115 – No. 15, April 2015, pp 21-26.

[4]. Suman, Seema Rani, Suresh Kumar, "Classification of Bug Reports Using Text Mining", International Journal of Advanced Research in Computer Science & Technology (IJARCST 2016), Issue 2 (Apr. - July 2016), pp 210-215.

[5]. Dr. A. R. Pon Periasamy A. Mishbahulhuda, "Data Mining Techniques in Software Defect Prediction", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 7, Issue 3, March 2017, pp 301-304.

[6]. Kirti Shamrao Tandale1, Prof. Bhavana Pansare,"A Survey on Effective Bug Triage with Data Reduction", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 12, December 2015, pp 12119-12125.

[7]. Haidar Osman, Mohammad Ghafari, Mircea Lungu, "An Extensive Analysis of Efficient Bug Prediction Configurations", ACM Conference PROMISE, November 8, 2017, pp 78-86.

[8]. Seyed Ali Asghar Mostafavi Sabet, Alireza Moniri,"Root-Cause and Defect Analysis based on a Fuzzy Data Mining Algorithm",(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 9, 2017, pp 21-29.

[9]. SEYED MOHAMMAD GHAFFARIAN and HAMID REZA SHAHRIARI,"Sofware Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques: A Survey", ACM Computing Surveys, Vol. 50, No. 4, Article 56. Publication date: August 2017, pp 56-92.

[10]. Yu Zhou1, Yanxiang Tong, Ruihang Gu1 and Harald Gall,"Combining text mining and data mining for bug report classification", JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS J. Softw. Evol. and Proc. 2016; 28:150–176.

[11]. Rafael Alcalá, María José Gacto, Jesús Alcalá-Fdez,"Evolutionary data mining and applications: A revision on the most cited papers from the last 10 years (2007–2017)",Journal of WIREs Data Mining Knowl Discov. Willey 2017, pp1-17.