International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 24 Issue 8 – MAY 2017.

ENHANCED BUG TRIAGE USING EFFECTIVE TERM SELECTION METHOD FOR A HIGH QUALITY BUG DATA SET

DEVALLA SATYA SOWJANYA [#1] and NAGARAJU TIRUKOTI [*2]

[#] PG Scholar, Kakinada Institute Of Engineering & Technology Department of Computer Science & Engineering , JNTUK,A.P, India.

[*] Assistant Prof, Dept of CSE, Kakinada Institute of Engineering & Technology, JNTUK, A.P, INDIA.

***Abstract*— In present scenario, large open source projects are available that guide to submit higher bug reports. The process of determining the errors is known as bug triage or bug assortments. Conventional works suffers from the issue of improper and irrelevant bug classification systems. In this paper, we propose an effective term selection method to categorize the bug data to its relevant cases. The intention of this study is to classify the bug reports into its relevant classes. We propose a term selection method that reduces the scale of the data as well as data quality. Experimental results prove the effectiveness of the systems.**

***Index Terms*— Bug triage, Open source projects, bug assortments, term selection and bug classification.**

I. INTRODUCTION

A recent development in the Open Source Technologies (OST) emancipates the bug repository which plays a significant role. A public bug repository generates a higher number of bugs under different classes. The software developer imposes various issues related to the software [1] that support possible suggestions to the bug reports. These enormous amounts of bug generation indulge the classification system more tedious. And also the entrepreneurs should spend more expenses to fix the bugs. Large scale data and low quality data are the two issues that affect the bug classification systems.

Data mining is considered as the promising solution to maintain the software data. By deploying data mining schemes, a bug in the bug repository is used to find the information and resolve the software problems. The discovered bugs are maintained in the bug repository [2]. Finding out the software bugs is an easy process whereas fixing the bugs is tedious and expensive process. In a bug repository, a bug is maintained as a bug report, which records the textual description of reproducing the bug and updates according to the status of bug fixing. A bug repository provides a data platform to support many types of tasks on bugs, e.g., fault prediction, bug localization, and reopened bug analysis. In this paper, bug reports in a bug repository are called bug data [3].

The rest of the paper is organized as follows: Section II describes the existing works in the field of bug assortments; Section III describes about the proposed framework; Section IV discusses about the experimental designs and at last concludes in Section V.

II. RELATED WORK

This section explains about the existing works in previous years. The author in [2] explored the bug data system using supervised machine learning systems. Naïve bayes classifier is used for classifying the bug system. The bug system is developed based on Eclipse.org which gives 30% classification accuracy. The author in [3] studied about 444, 500 reports from Mozilla and Eclipse projects. The time taken for classifying the bugs is in higher rate. They discovered some tossing graph from the bug history. By doing so, the accuracy rate of 72% is achieved.

The author in [4] studied about the novel technique, named, COSTRIAGE. The experiment reduces the cost without significantly sacrificing accuracy. They used a proof-of-concept implementation by using cost of bug fixing time. Developer profile model is general enough to support other code indicators such as interests, efforts, and expertise to optimize for both accuracy and cost for automatic bug triage. The author in [8] investigated the duplicate records of the bug data. They proposed two schemes, BM25F and BM25Fext, which used for the similarity measure. The experimental results prove that BM25F recall the rate by 13% than the BM25F.

The author in [7] demonstrated about project specific bug finding tool. The tool works on the basis of memories of bug fixes. The bugs are detected by BugMem and it is fixed. Some source code repositories are stored in CVS and Subversion of the system. They solved the information to enhance source code quality in detailed manner. Then it is further enhanced by reducing the scale data sets and better feature selection. This takes attributes of each bug data set and train a predictive model based on historical data sets. For experiments they use bug data set of Eclipse and Mozilla and get high quality bug data set [8].

III. PROPOSED FRAMEWORK

The proposed framework is categorized into five phases. It is listed as follows:

i) ***Dataset Collection:*** The data is collected from bug repository that contains activities, results, context and other factors. The data collection plays a vital role to evaluate the classification. The data is stored in the form of statistical matrix where each column represents a specific variable.

ii) ***Preprocessing methods:*** Data preprocessing is known as data cleaning process. It is processed by resolving missing values, smoothing the noisy data and resolving the inconsistencies. The irrelevant data is also eliminated from the dataset.

iii) ***Feature selection and instance selection:*** The integration of instance selection and feature selection is employed to reduce the scale of the data. The task of instance selection is to reduce the number of instances by eliminating the noise and redundant sets. By disposing the high dimensional data, an effective bug triage is obtained.

iv) ***Bug data reduction:*** The task is to reduce the level of bug data. It also used for improving the accuracy of the bug triage. Concurrently it also reduces the scale and quality of the data.

v) ***Performance evaluation:*** The algorithm is used for reduced the high dimensional spaces using some non-representative instances. The quality of bug triage can be measured with the accuracy of bug triage to reduce noise and redundancy in bug data sets.

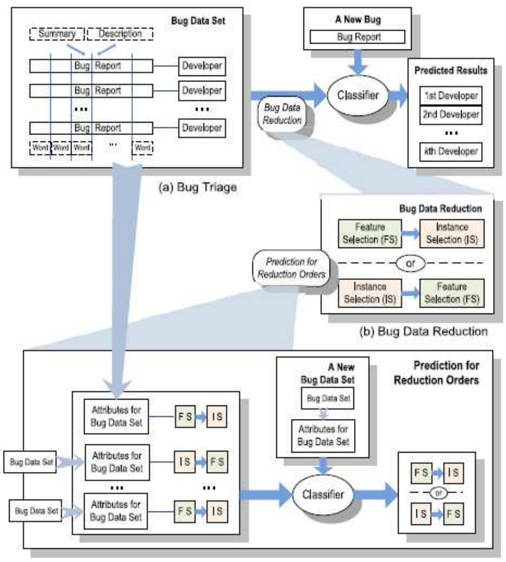

Fig.3.1. System Architecture

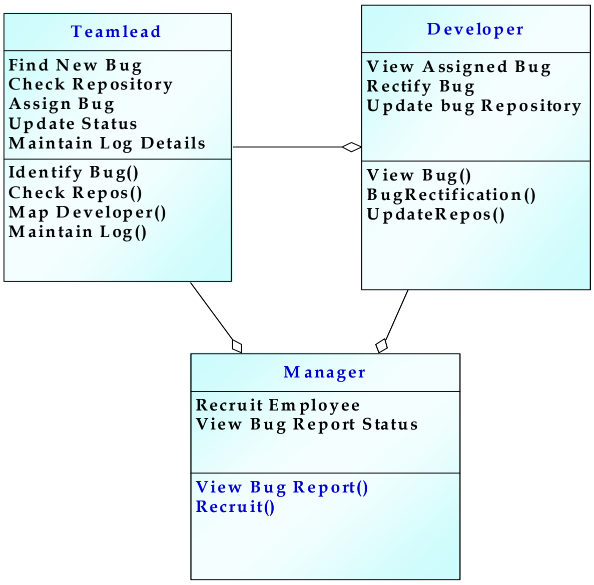

Fig. 3.2 Class diagram representing the task of each actor

IV. EXPERIMENTAL DESIGNS

This section presents the experimental designs of the bug triage system. It is explained as follows:

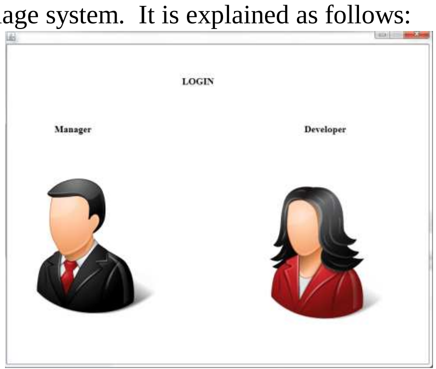

Fig.4.1 Bug Triage System- Login page of the system

Fig.4.2 Manager's login

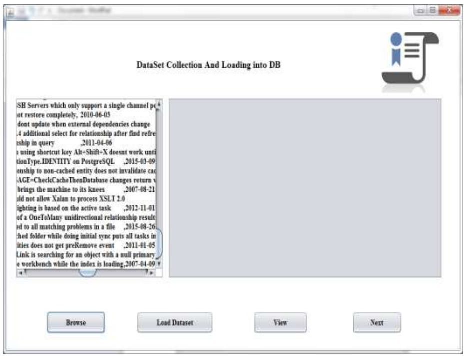

Fig.4.3 loading the bug dataset

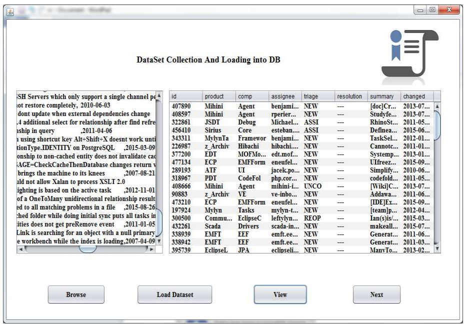

Fig.4.4 Arranging the dataset into its attributes

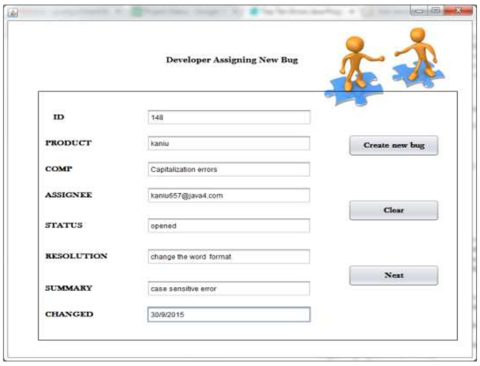

Fig.4.5 registering new bug to the systems

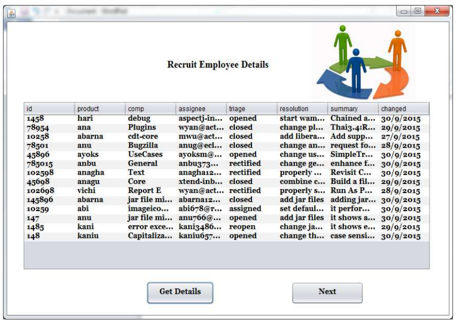

Fig.4.6 Viewing the bugs posted by different developers

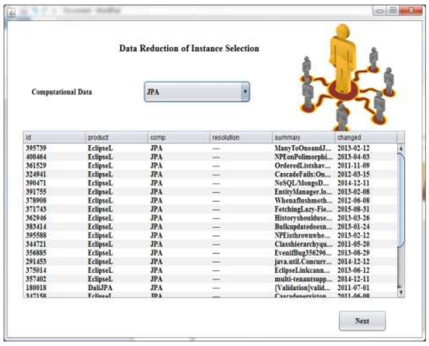

Fig. 4.7 Viewing the bugs using java instances

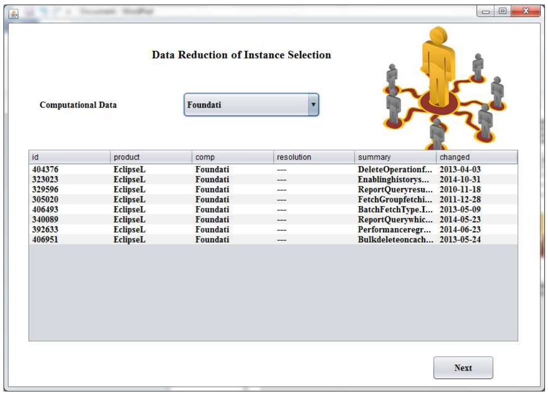

Fig.4.8 Viewing the bugs using foundation instances

Fig. 4.9 Bug report analysis is done via three classifications namely, opened, assigned and closed.

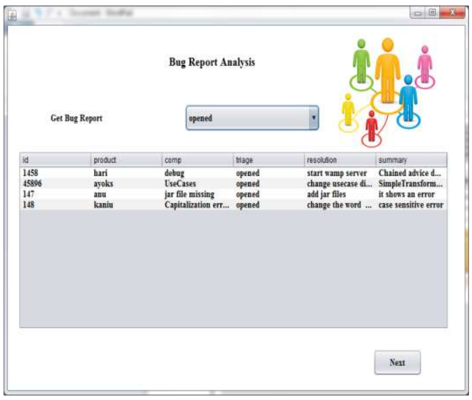

Fig.4.10 Opened classification systems

Fig.4.11 closed classification system

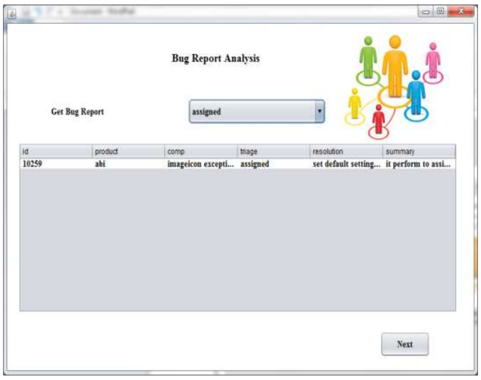

Fig.4.12 Assigned classification system

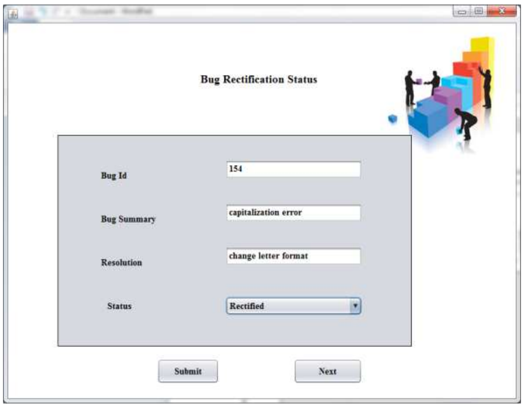

Fig.4.13 Viewing the status of the bug rectification

Fig.4.14 Efficiency of the bug triage systems

V. CONCLUSION

In this paper we have focused on reducing bug data set in order to have less scale of data and quality data. For that we have used feature selection and instance selection techniques of data mining as well as we have used historical data. Our experimental results showed that this data reduction technique will give quality data as well as it will reduce the data scale. We have added new module in this paper than the earlier which will give various details related to the bugs to administrator in graphical format. In future work, we plan on improving the results of data reduction more in bug triage to explore how to prepare a high quality bug data set.

REFERENCES

[1] Jifeng Xuan, He Jiang, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu," Towards Effective Bug Triage with Software Data Reduction Techniques" ieee transactions on knowledge and data engineering, vol. 27, no. 1, january 2015

[2] Mamdouh Alenezi and Kenneth Magel, Shadi Banitaan "Efficient Bug Triaging Using Text Mining" © 2013 academy publisher

[3] Francisco Servant "Supporting Bug Investigation using History Analysis" 978-1-4799-0215-6/13 c 2013 IEEE

[4] Pamela Bhattacharya, Iulian Neamtiu, Christian R. Shelton, "Automated, Highly- Accurate, Bug Assignment Using Machine Learning and Tossing Graphs", May 2, 2012

[5] K. Balog, L. Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise corpora," in Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, Aug. 2006, pp. 43–50.

[6] P. S. Bishnu and V. Bhattacherjee, "Software fault prediction using quad tree-based k-means clustering algorithm," IEEE Trans. Knowl. Data Eng., vol. 24, no. 6, pp. 1146–1150, Jun. 2012.

[7] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," Data Mining Knowl. Discovery, vol. 6, no. 2, pp. 153–172, Apr. 2002. A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," Knowledge-Based Systems, vol. 36, no. 0, pp. 226–235, 2012.

[8] S. Kim, H. Zhang, R. Wu, and L. Gong, "Dealing with noise in defect prediction," in Proc. 32nd ACM/IEEE Int. Conf. Softw. Eng., May 2010, pp. 481–490.

[9] A. Lamkanfi, S. Demeyer, E. Giger, and B. Goethals, "Predicting the severity of a reported bug," in Proc. 7th IEEE Working Conf. Mining Softw. Repositories, May 2010, pp. 1–10.

[10] A. Lamkanfi, S. Demeyer, E. Giger, and B. Goethals, "Predicting the severity of a reported bug," in Proc. 7th IEEE Working Conf. Mining Softw. Repositories, May 2010, pp. 1–10.

AUTHOR PROFILE:

DEVALLA SATYA SOWJANYA is a student of Kakinada Institute Of Engineering & Technology affiliated to JNTUK, Kakinada pursuing M.Tech (Software Engineering). Her Area of interest includes Data Mining and its objectives in all current trends and techniques in Computer Science.

NAGARAJU TIRUKOTI M.TECH is working as Assistant Professor, Department of Computer Science & Engineering, Kakinada Institute of Engineering & Technology, JNTUK, A.P, INDIA.