Bug Triage with Enhanced Data Reduction using PSO

Ms.G.Kousalya[1], Mr.S.J.Vivekanandan[2]
PG Scholar[1], Research Scholar[2]
[1]M.Tech., Dept. of Computer Science & Engineering, PRIST University,Vallam ,Thanjavur.
[2]Assistant Professor, Dept. of Computer Science & Engineering,PRIST University,Vallam ,Thanjavur.

***Abstract*— Large open source projects receive copious rates of submitted bug reports. Triaging these incoming bug reports manually is difficult and consumes more time. The purpose of bug triage is to assign possible experienced developers to new coming bug report. In proposed approach we are using the text classification techniques to perform automatic bug triage. We are using instance selection and feature selection at the same time with the attributes of the historical bug data. It reduces the problem in data reduction, i.e., reduces the size of bug data and enhance the eminence of the bug data. PSO method is used to predict the expert for fixing the new bug and status about the bug report is also updated whether it is assigned to any developer or not and it is rectified or not.**

***Index Terms*— Bug, Bug triage, data reduction, Instance selection, Data Mining.**

I. INTRODUCTION

Data mining plays an important role in retrieving information from the large data storage. Open source projects in software companies maintain software repositories to record all the information about the development of the project. For open source large-scale software projects, the number of daily bugs is so large which makes the triaging process very difficult and challenging.

Manuscript received March, 2016.

***Ms.G.Kousalya**, M.Tech, Department of Computer Science and Engineering, PRIST University Thanjavur, India.*

***Mr.S.J.Vivekanandan**, Research scholar, Assistant professor, Department of Computer Science and Engineering, PRIST University Thanjavur, India.*

There are two challenges related to bug data that may have an effect on the effective use of bug repositories in software Development tasks, namely the large scale and the low quality. In a bug storage area, a bug is maintained as a bug report, which report the textual description of reproducing the bug and updates the status of bug fixing.

Dealing with more number of software bug reports in the bug repository, data mining is performed to handle software engineering problems. The ultimate aim of the tester is to report about the bug clearly and it is maintained as a report in the bug repository. Bug triage is a process where tester issues are spotted and prioritized. During bug triaging software development team should ensure about the required time to fix the bug. Allocating a bug to the specified developer if fails, then the overall cost increase and the product release date also is postponed. So predicting the expert for fixing the bug is essential in bug triage.

Here automatic approach is used rather using manual triage; it reduces the expensive in time cost and low in accuracy level. Like existing method, text classification method is used [1]. In proposed system data reduction problem is to be reduced by applying instance selection and feature selection at the same time in the bug data set. PSO (particle swarm optimizer) method is initialized with the group of solutions and then searches for best possible by updating solutions. Particle move through the solution space and select the fitness criterion for each iteration by following the best values it has achieved so far. Every time the solution is updated in the bug report clearly.

With that solution, the bug report is fixed by an expert and the status about the bug report is updated. Here in this paper we are using feature selection and instance selection along with historical data for reducing the bug data in bug

repository. Thus diminish the size of the data and exceed the quality of the bug data.

II. ARCHITECTURE

a) Aim of bug triage is to assign a developer for bug fixing. Once an expert is assigned to a new bug report he will fix the bug or try to rectify it. He will give the status related to bug whether it is rectified or not [2].

b) Data Reduction

Here we are reducing the bug data by using instance and feature selection so that we get low scale as well as quality data.

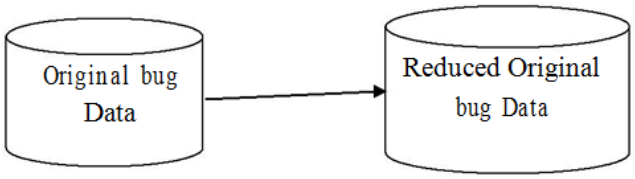

Figure 1: Reduced Data Set

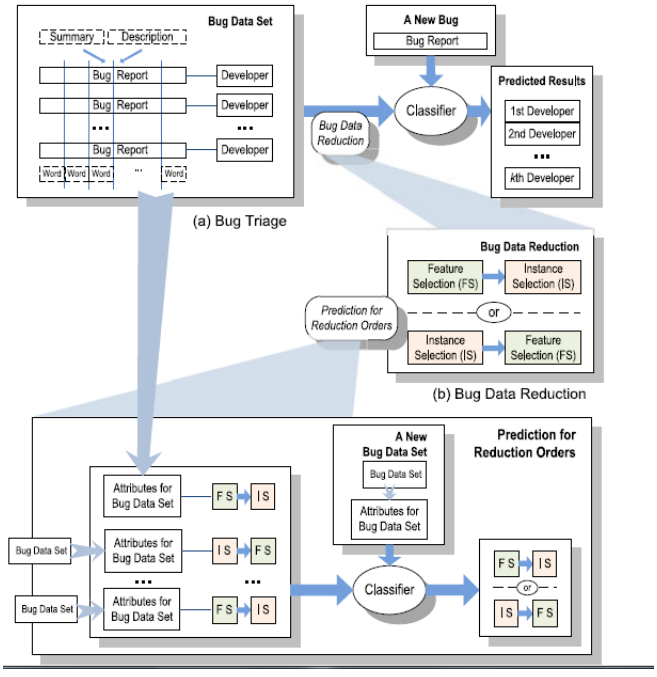

Figure 2: Process of Bug Triage

c) Classifier

In this section, original data set is classified with the upcoming new bug report. Here we are reducing the bug data by using instance and feature selection so that we get low scale as well as quality data [1]. Then the priority list of the developers is maintained according to their performance in solving bug report[9] . Once an expert solved very few bugs and remains idle then the expert is removed from the priority list.

III. IMPLEMENTATION

There are two approaches in mining the bug reports in the bug storage. Feature selection method is used first to reduce words and Instance selection is used to remove repeated and blank reports. Thirdly, PSO approach is used to find the expert for fixing the bug.

Algorithm 1. Data reduction based on FS → IS

Input: training set $\mathcal{T}$ with n words and m bug reports,
reduction order FS→IS
final number n_F of words,
final number m_I of bug reports,
Output: reduced data set $\mathcal{T}_{FI}$ for bug triage

1) apply FS to n words of $\mathcal{T}$ and calculate objective values for all the words;
2) select the top n_F words of $\mathcal{T}$ and generate a training set $\mathcal{T}_F$;
3) apply IS to m_I bug reports of $\mathcal{T}_F$;
4) terminate IS when the number of bug reports is equal to or less than m_I and generate the final training set $\mathcal{T}_{FI}$.

The algorithm depicts the overall data reduction method in it. Existing bug reports in the bug storage treated as training set and the new coming bugs are the test set.

1. FEATURE SELECTION

It selects a minimum set of features such that the prospect distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features[4][5]. Reduce specified number of patterns in the patterns, easier to understand. Select new attributes in the data set that can detain the key information in a data set much more proficiently than the unique attributes. Use the smallest depiction which is enough to solve the task [3].

2. INSTANCE SELECTION

Instance selection method is associated with the classification and clustering methods in data mining. It is not easy to find the valid one without the patterns in the bug report. The attributes of the data are taken and it is classified according to their patterns. It is independent and takes the original data set attributes and finds the reduced data set. Classifier is used to classify the attributes in every bug

report.

3. PARTICLE SWARM OPTIMIZER

PSO is initialized with the group of solutions and then searches for best possible by updating solutions. Particles move through the solution space and select the fitness criterion from each iteration by following the best values it has achieved so far. With that solution, the bug report is fixed by the expert.

IV. MODULE DESCRIPTION

This module show's four part's as follow:

1. Firstly it will show how many bugs are not assigned to any developer. It will give complete status about the bugs to the admin so that he will come to know which bugs are not assigned yet.
2. Secondly it will show how many bugs are not assigned to any developer. It will give complete status about the bugs to the reporter so that he will come to know which bugs are assigned.
3. Thirdly it will show how many bugs are rectified by the developer's. It will give complete status about the bugs to the admin so that he will come to know which bugs are rectified completely.
4. Fourthly it results the number of bugs are not fixed by the developer's. It will give complete status about the bugs to the admin so that he will come to know which bugs are not rectified yet. Historical data also used for reducing the bug data. Here we selected the recent accessed bugs and the data which we get we will use it for data reduction.

V.EXPERIMENTAL RESULTS

This section illustrates the order of applying the best approach in data reduction. First the set of data sets are collected from large projects Eclipse and Mozilla. Each bug has its own attributes in the bug data set. Some of the attributes are lines of codes, intelligence, difficulty, number of operands, severity[10], priorities, unique words, reporters, experts, product, number of operators, and etc.,

	Name	DS-E1	DS-E2	DS-E3	DS-E4	DS-E5
Eclipse	Range of Bug IDs	200001 - 220000	220001 - 240000	240001 - 260000	260001 - 280000	280001 - 300000
	# Bug reports	11,313	11,788	11,495	11,401	10,404
	# Words	38,650	39,495	38,743	38,772	39,333
	# Developers	266	266	286	260	256
	Name	DS-M1	DS-M2	DS-M3	DS-M4	DS-M5
Mozilla	Range of Bug IDs	400001 - 440000	440001 - 480000	480001 - 520000	520001 - 560000	560001 - 600000
	# Bug reports	14,659	14,746	16,479	15,483	17,501
	# Words	39,749	39,113	39,610	40,148	41,577
	# Developers	202	211	239	242	273

Table 1: Sample Data Sets from Eclipse and Mozilla

Instance selection method first in the collected data set and secondly feature selection method is used. When performing these both methods in the gathered data set, more number of unnecessary and frequent bug reports is removed from the selected data set. With that collected data set, expert is to be searched by using the Particle Swarm Optimizer approach is used to select the developer from the best solution giver list (expert) and the bug is fixed.

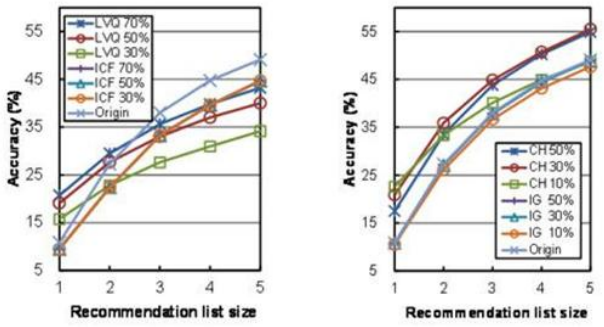

Figure 3: The List of Developers by Instance Selection in Mozilla and Feature Selection in Mozilla

Finally, the status of the bug report shown in the bar chart depicts the four categories of not assigned, assign, rectified and not rectified.

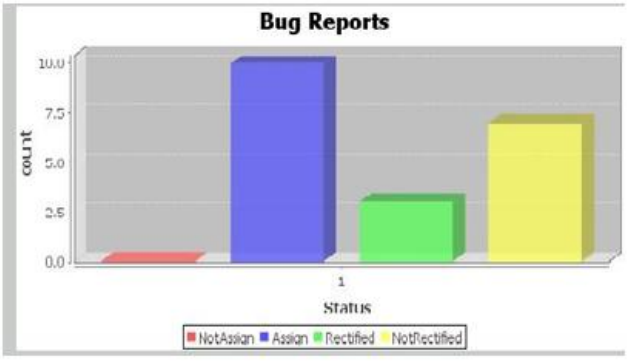

Figure 4: Status of the Bug Report

VI. CONCLUSION

In this paper we have focused on reducing the problem in data reduction and assigning the appropriate developer to the new bug. For that we have examined the historical bug reports as training set and the new bugs are treated as test set. By using the reduction order, instance selection and feature selection at the same time, duplicated bug reports and noisy words are removed .It reduces the size of the data and improve the eminence of the bug data.

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. Using PSO method is used to find the expert to fix the bug correctly. In future work, we plan on enhance the results of data reduction efficient in bug triage to explore how to prepare a high quality bug data set.

REFERENCES

1) "Towards Effective Bug Triage with Software Data Reduction Techniques" Fifing Xian, He Jiang, Member, IEEE, Yan Hub, Shirley Ran, Wailing Zou,Zhongxuan Luo, and Xindong Wu, Fellow, IEEE.
2) Bugzilla, (2014). [Online]. Avaialble: http://bugzilla.org/
3) C. C. Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform. Syst., vol. 36, no. 1, pp. 1–21, 2013.
4) V. Bolon-Canedo, N. Sanchez-Maro~no, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," Knowl. Inform. Syst., vol. 34, no. 3, pp. 483–519, 2013.
5) A. K. Farahat, A. Ghodsi, M. S. Kamel, "Efficient greedy feature selection for unsupervised learning," Knowl. Inform. Syst., vol. 35, no. 2, pp. 285–310, May 2013.
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) J. Anvik and G. C. Murphy, "Reducing the effort of bug report triage: Recommenders for development-oriented decisions," ACM Trans. Soft. Eng. Methodol., vol. 20, no. 3, article 10, Aug. 2011.
8) S. Artzi, A. Kie_zun, J. Dolby, F. Tip, D. Dig, A. Paradkar, and M. D. Ernst, "Finding bugs in web applications using dynamic test generation and explicit-state model checking," IEEE Softw., vol. 36, no. 4, pp. 474–494, Jul./Aug. 2010.
9) J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?" in Proc. 28th Int. Conf. Softw. Eng., May 2006, pp. 361–370.
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

[1]G.KOUSALYA completed B.Tech., in Information Technology from Parisutham Institute of Technology and Science, Anna University in 2013.She is pursuing her M.Tech., Computer Science and Engineering in PRIST University, Thanjavur. Her research interests are Data Mining, Big Data, and Database.

[2]S.J.VIVEKANANDAN pursuing PhD in Sathyabama University Chennai, Tamilnadu. He received B.Tech., Information Technology and Management from SASTRA University in 2008 and M. Tech., Computer Science and Engineering from SASTRA University in 2010.He is currently working as Assistant Professor in PRIST University, Thanjavur. His research interests are Data Mining , Utility Mining, Data Structures and Database.

