ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Visit: www.ijirset.com*

Vol. 7, Issue 3, March 2018

# Bug Traige with Data Reduction Techniques

Shubhada Kamble, Umeshpedde, Swati Londhe, PoonamVaidya, Prof. S. A. Mule,

B. E Student, Dept. of IT, PVGCOET, Pune, India

B. E Student, Dept. of IT, PVGCOET, Pune, India

B. E Student, Dept. of IT, PVGCOET, Pune, India

B. E Student, Dept. of IT, PVGCOET, Pune, India

Guide, PVGCOET, Pune, India

***Abstract*:** Bugs are one of the critical issues for any software firm.Most of the softwarefirms pay near about 45 percent value to manage these bugs.An inevitable step of managing bugs isbug triage, which involvesassigning new incoming bug to an expert person who can solve this bug.To handle these bugs manually is very time consuming process.To decrease the time in manual work, text classification techniques are applied to conduct automatic bug triage.This paper address the matter of reduction for bug triage, i.e., the way to reduced scale and improve the standard of bug information. We combine instance selection method with feature selection method to reduced information scale on the bug dimension and the word dimension.Then we propose a binary classification method to predict the order of instance selection and feature selection method based on the attributes of historical bug datasets.This method of information reduction will effectively reduce the scale of dataset and improve the accuracy of bug triage technique.Then the bugs are distributed to bug solving experts.

KEYWORDS: Bug data reduction,feature selection, instance selection, bug triage, prediction for reduction orders

## I. INTRODUCTION

Many of the software companies need to deal with large amount of software bugs daily. Software bugs are unavoidable and fixing these software bugs is a very expensive task. In fact, many of the Software organization spend lots of their resources in managing these bugs. For handling this bugs, bug repositories are used .Bugs which are reported by users are stored in the bug repository. In a bug repository, a bug is maintained as a bugreport, which records the textual description of reproducing the bug and updates according to the status of bug fixing[23]. Many open source software projects have an open bug repository that permits both users and developers to submit faults or issues in the software and suggest possible improvements.For open source large scale software projects, the number of daily bugs is so large which makes the triaging process very difficult and challenging. There are two challenges related to bug data that may affect the effective use of bug repositories in software development tasks, namely the large scale and the low quality.
When a bug report is formed, a human triage is used to provide this bug to a developer, who will try to fix these bugs. Bug assignment to the developer is also recorded in a bug report. The method of assigning a correct developer for fixing the bug is known as bug triage. Bug triage basically is one of the most time consuming step in managing of bugs in software projects. Manual bugtriage is very time consuming process.In traditional bug repositories system, all the large amount of bugs were manually triaged by somespecialized developersi.e human triager.Manual bug triage is expensive in time cost and low in accuracy because of large number of daily bugs. To avoid this problem existing system has proposed automatic bug triage approach [1]. This approach applies text classification techniques to predict the developer for bug report. Based on the result of text classification bug is assigned to specialized developer.To improve the result of text classification techniques for bug triage, some further techniques are investigated, e.g., a tossing graph approach [10] and a collaborative filtering approach[13].Large-scale and low-quality bug data in bug repositories block the techniques of automatic bug triage.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Visit: www.ijirset.com*

**Vol. 7, Issue 3, March 2018**

This paper proposes data reduction technique for bug triage to reduce the size of bug dataset and improve the quality of bug dataset. Data reduction techniques is used to remove some bug reports and words which are redundant and non informative. For that we used two methods i.e instance selection(IS) and feature selection(FS).The order of applying instance selection and feature selection algorithms may also affect the bug triage process. So to decide order, we build a binary classifier based on the historical data set. This process gives us reduced data set that can be usedfor assigning developer toan incoming bug .

## II.LITERATURE SURVEY

Bug triage aims to assign an appropriate developer to fix anew bug, i.e., to determine who should fix a bug. Cubrani_cand Murphy [5] first proposed the problem of automatic bugtriage to reduce the cost of manual bug triage. They appliedtext classification techniques to predict related developers for new incoming bug.In their work a bug report is mapped to a document and anassigned developer is mapped to the label of the document.Then, bug triage is converted into a problem of text classificationandthen it is automatically solved with the help of any mature text classificationtechniques, for e.g., Naive Bayes [5]. Based on the results a human triager assigns newbugs by incorporating his/her expertise.

Anvik andMurphy [2] extended above work to reduce the effort of bug triageby creating development-oriented recommenders.

Jeong et al. [10] found that over 37 percent of bug reports had been reassigned in manual bug triage.They introduced a graph model,which captures bug tossing history. This graphmodel reveals developer networks which can be used to discover team structures and to find suitable experts for a new task. Then it helps to assign developers to bug reports, correctly.

To avoid low-quality bug reports in bug triage,Xuan et al. [19] trained a semi-supervised classifier by combiningunlabeled bug reports with labeled ones.This new approach combines naive Bayes classifier and expectation-maximization to take advantage of both labeled and unlabeled bug reports. This approach trains a classifier with a fraction of labeled bug reports. Then the approach iteratively labels numerous unlabeled bug reports and trains a new classifier with labels of all the bug reports. Theyalso employed a weighted recommendation list to boost the performance by imposing the weights of multiple developers in training the classifier.

Park et al. [13]converted bug triage into an optimization problem and proposea collaborative filtering approach to reduce the bugfixingtime.In a general recommendation problem,content based recommendation (CBR)is well known to suffer from over specializationrecommending only the types of bugs that each developer has solved before. This problem is critical in practice, as some experienced developers could be overloaded, and this would slow the bug fixing process. Then they took two directions to address this problem: First, they reformulated the problem as an optimization problem of both accuracy and cost. Second, they adopted a content-boosted collaborative filtering (CBCF), combining an existing CBR with a collaborative filtering recommender (CF), which enhances the recommendation quality of either approach alone.

## III. PROPOSED SYSTEM

This paper address the issue of information reduction for bug triage, i.e., how to decrease the bug information to save the work cost of designers and enhance the quality to encourage the procedure of bug triage. Information reduction for bug triage plans to construct a little scale and great arrangement of bug information by evacuating bug reports and words, which are excess or non-instructive.Fig.1 shows architecture of the system.Before training a classifier with a bug data set a phase of data reduction will added which combines the techniques of instance selection and feature selection to reduce the scale of bug data. The order of applying instance selection algorithm and a feature selection algorithm may affect the results of bug triage. FS->IS and IS->FS are viewed as two orders for applying reducing techniques. To avoid the time cost of manually checking both reduction orders, historical data set is used to predict the reduction orders.This paper propose a predictive model to determine the order of applying instance selection and feature selection.IS and FS techniques are applied according to that order. After applying these two techniques reduced

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 3, March 2018

dataset is generated. Now this reduced dataset is used to train a classifier for predicting assignment of developer for new bug.

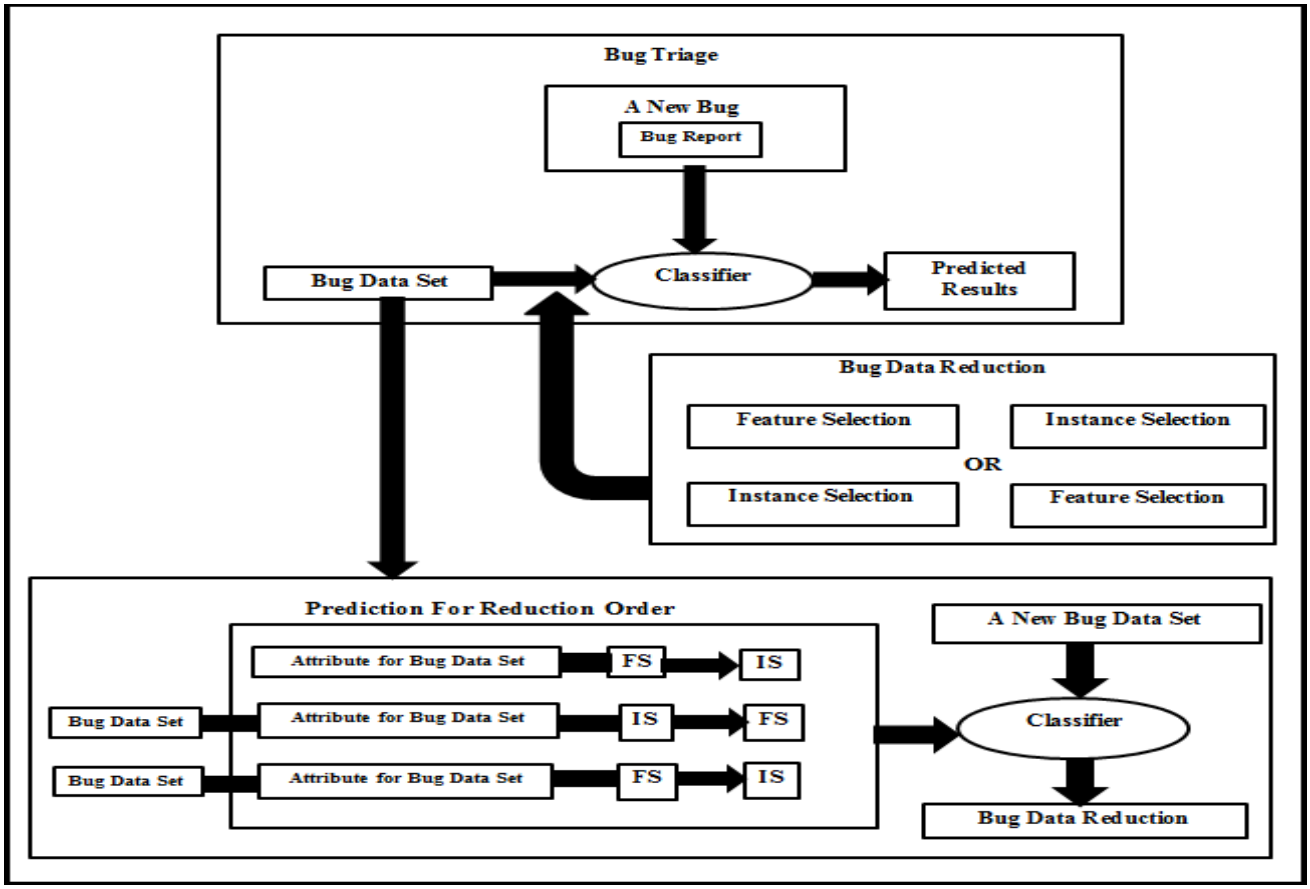


Fig.1.Architecture of our system

**3.1Data Reduction:**

We combine techniques of instance selection and feature selection to remove uninformative bug reports and words.

Instance selection :Instance selection technique is used to obtain a subset of relevant instances. Noisy and redundant bug reports are removed in instance selection[4]. An instance selection algorithm can provide a reduced data set by removing non-representative instance[12]. There are four instance selection algorithms namely Iterative Case Filter (ICF)[3],Learning Vector Quantization (LVQ)[11] , Decremental Reduction Optimization Procedure (DROP)[22] , and Patterns by Ordered Projections(POP)[14].Among these algorithm we will use Iterative Case Filter (ICF) as it provides better accuracy.

Feature selection: Feature selection is the process of selecting a subset of relevant features[6]. Feature selection improves the accuracy of bug triage by removing uninformative words.There are four algorithns of Feature selection namely Information Gain (IG)[9] , x2 statistic (CH)[20],Symmetrical Uncertainty attribute evaluation(SU)[18], and Relief-F Attribute selection (RF)[15].Among these algorithm we will use x2 statistic (CH)as it provides better accuracy.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 3, March 2018

**3.2 Prediction for Reduction Orders**

Given an instance selection algorithm IS and a feature selection algorithm FS, FS ->IS and IS ->FS are viewed as two orders of bug data reduction.To avoid the time cost of manually checking accuracy of both reduction orders, we proposea predictive model to determine the order of applying instance selection and feature selection. To build a binary classifier to predict reduction orders, we extract the attributes from historical bug data sets. Then, we train a binary classifier on bug data sets with extracted attributes and predict the order of applying instance selection and feature selection for a new bug data set. We employ the classifier AdaBoost to predictreduction orders since AdaBoost is useful to classify imbalanced data and generates understandable results of classification.

**Algorithm for data reduction:**

FS → IS = Bug data reduction. Which first applies FS and then IS .On the other hand, IS → FS denotes first applying IS and then FSIn Algorithm 1, we briefly present how to reduce the bugdata based on FS → IS. Given a bug data set, the output ofbug data reduction is a new and reduced data set.

**Algorithm 1 : Data reduction based on FS -> IS**

**Input :** Training set *T* with n words and m bug reports
Reduction order FS ->IS
Final number *nf*of words ,
Final number $m_I$ bug reports,

**Output:**reduced data set $T_{FI}$ for bug triage

1) Apply FS to n words of *T* and calculate objective values for all the words.
2) Select the top*nf* words of *T* and generate a training set *Tf* ;
3) Apply IS to $m_I$ bug report of *T f ;*
4) Training IS when the number of bug report is equal to or less then $m_I$ and generate to the final training set $T_{FI}$ ;

Two algorithms FS and IS are applied sequentiallyIn Step 2), some of bug reports may be blank during feature SelectionIn our work, FS → IS and IS → FS are viewed as two orders of bug data reduction.

## IV. CONCLUSION

Bug fixing is an elegant part of software organization. One of the major challenges in process of bug triage is to assign a skilled developer to fix a new bug. This paper proposes the complete abstraction of an automatic bug triage approach and a framework that removes the issue of bug data to a huge extent. Feature selection and Instance selection techniques are combined to obtain better quality of bug data. Use of NBClassifier is proposed for suggesting a list of expert developers for fixing the bug.

## V. ACKNOWLEDGMENT

Authors want to acknowledge Principal, Head of department and guide of their project for all the support and help rendered. To express profound feeling of appreciation to their regarded guardians for giving the motivation required to the finishing of paper.

## REFERENCES

[1] J. Anvik, L. Hiew, and G. C. Murphy, “Who should fix this bug?”inProc. 28th Int. Conf. Softw. Eng., May 2006, pp. 361–370.
[2] J. Anvik and G. C. Murphy, “Reducing the effort of bug report triage:Recommenders for development-oriented decisions,” ACMTrans. Soft. Eng. Methodol., vol. 20, no. 3, article 10, Aug. 2011.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Visit: www.ijirset.com*

**Vol. 7, Issue 3, March 2018**

[3] 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.
[4] V. Cerver_on and F. J. Ferri, "Another move toward the minimum consistent subset: A tabu search approach to the condensed nearest neighbor rule," IEEE Trans. Syst., Man, Cybern., Part B, Cybern., vol. 31, no. 3, pp. 408–413, Jun. 2001.
[5] D. _Cubrani_c and G. C. Murphy, "Automatic bug triage using text categorization," in Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng., Jun. 2004, pp. 92–97.
[6] 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.
. [7] I. Guyon and A. Elisseeff, "An introduction to variable and featureselection," J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003.
.[8] A. E. Hassan, "The road ahead for mining software repositories," inProc. Front. Softw. Maintenance,
Sep. 2008, pp. 48–57.
[9] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques,3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.
[10] G. Jeong, S. Kim, and T. Zimmermann, "Improving bug triagewith tossing graphs," in Proc. Joint Meeting 12th Eur. Softw.Eng.Conf. 17th ACM SIGSOFT Symp. Found. Softw. Eng., Aug. 2009,pp. 111–120.
[11] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K.Torkkola, "LVQ_PAK: The learning vector quantization programpackage," Helsinki Univ. Technol., Esbo, Finland, Tech.Rep. A30, 1996.
[12] J. A. Olvera-L_opez, J. F. Mart_ınez-Trinidad, and J. A. Carrasco-Ochoa, "Restricted sequential floating search applied to objectselection," in Proc. Int. Conf. Mach. Learn. Data Mining Pattern Recognit.,2007, pp. 694–702.
[13] J. W. Park, M. W. Lee, J. Kim, S. W. Hwang, and S. Kim,"Costriage: A cost-aware triage algorithm for bug reporting systems,"inProc. 25th Conf. Artif.Intell., Aug. 2011, pp. 139–144.
[14] J. C. Riquelme, J. S. Aguilar-Ru_ız, and M. Toro, "Finding representativepatterns with ordered projections," Pattern Recognit., vol. 36,pp. 1009–1018, 2003.
[15] M. Robnik-_Sikonja and I. Kononenko, "Theoretical and empiricalanalysis of relieff and rrelieff," Mach. Learn., vol. 53, no. 1/2,pp. 23–69, Oct. 2003.
[16] S. Shivaji, E. J. Whitehead, Jr., R. Akella, and S. Kim, "Reducingfeatures to improve code change based bug prediction," IEEETrans.Soft. Eng., vol. 39, no. 4, pp. 552–569, Apr. 2013.
[17] J. Tang, J. Zhang, R. Jin, Z. Yang, K. Cai, L. Zhang, and Z. Su,"Topic level expertise search over heterogeneous networks,"Mach. Learn., vol. 82, no. 2, pp. 211–237, Feb. 2011.
[18] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: PracticalMachine Learning Tools and Techniques, 3rd ed. Burlington, MA,USA: Morgan Kaufmann, 2011.
[19] J. Xuan, H. Jiang, Z. Ren, J. Yan, and Z. Luo, "Automatic bug triageusing semi-supervised text classification," in Proc. 22nd Int.Conf. Softw. Eng. Knowl. Eng., Jul. 2010, pp. 209–214.
[20] Y. Yang and J. Pedersen, "A comparative study on feature selectionin text categorization," in Proc. Int. Conf. Mach. Learn., 1997,pp. 412–420.
[21] H. Zhang and G. Sun, "Optimal reference subset selection fornearest neighbor classification by tabu search," Pattern Recognit.,vol. 35 pp. 1481–1490, 2002.
[22] D. R. Wilson and T. R. Mart_ınez, "Reduction techniques forinstance-based learning algorithms," Mach. Learn., vol. 38,pp. 257–286, 2000.
[23] T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schr€oter, and C. Weiss, "What makes a good bug report?" IEEE Trans.Softw. Eng., vol. 36, no. 5, pp. 618–643, Oct. 2010.