ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

An Effective and Accurate Bug Triage System with Software Data Reduction Techniques

Varsha Bongane [1], S.B.Natikar [2]

P.G. Student, Department of Computer Engineering, VACOE, Ahmednagar, Maharashtra, India[1]

Associate Professor, Department of Computer Engineering, VACOE, Ahmednagar, Maharashtra, India[2]

ABSTRACT: The procedure of altering bug will be bug triage, which expects to effectively allot a designer to another bug. Programming organizations spend the greater part of their expense in managing these bugs. To decrease time and cost of bug triaging, we introduce a programmed way to deal with anticipate an engineer with important experience to explain the new coming report. In proposed approach we are doing information diminishment on bug information set which will decrease the size of the information and in addition build the nature of the information. We are utilizing occasion determination and highlight choice all the while with verifiable bug information. We have included another module here which will portray the status of the bug such as whether it relegated to any designer or not and it is redressed or not.

KEYWORDS: Mining software repositories, bug, bug triage, data reduction, bug data reduction, feature selection, instance selection, bug triage, prediction for reduction orders.

I. INTRODUCTION

A Bug storehouse assumes an essential part in overseeing programming bugs. Numerous open source programming ventures have an open bug archive that permits both engineers and clients to submit deserts or issues in the product, propose conceivable improvements, and remark on existing bug reports. For open source extensive scale programming extends, the quantity of every day bugs is so expensive which makes the triaging handle extremely troublesome and testing [2]. Programming organizations spend more than 45 percent of expense in settling bugs. There are two difficulties identified with bug information that might influence the viable utilization of bug vaults in programming improvement assignments, in particular the substantial scale and the low quality. In a bug storehouse, a bug is kept up as a bug report, which records the printed depiction of recreating the bug and upgrades as indicated by the status of bug altering [1].

Bug settling is a noteworthy and tedious procedure in programming support [3]. For a vast scale programming extend, the quantity of day by day bugs is large to the point that it is difficult to handle them without deferring [2]. The work of overseeing bugs expands the expense of programming quality upkeep [15]. Numerous product ventures utilize a bug following framework to store and oversee bugs presented by clients, including end clients, analyzers, and designers [1]. The bug following framework gives a stage, where clients can speak with one another amid the bug settling process. Bugzilla1 is such a bug following framework, which is utilized by numerous huge open source programming ventures. In light of the bug following framework, the engineers can without much of a stretch pursuit and keep up all the current bugs. Bug triage, an imperative stride for bug settling, is to appoint another bug to a significant designer for further taking care of [3]. A general technique for bug triage is to appoint bugs physically.

Practically speaking, because of the successive changes of programming advancement groups, it is hard to distinguish the right engineer in manual triage [3]. Taking Eclipse 2 as a sample, Anvik reports that a normal of 37 bugs for each day are submitted to the bug following framework and 3 man hours for every day are required for the manual triage [2]; the exact study by Jeong et al. demonstrates that 44 percent of bugs have been allotted to the wrong designer after the primary task [4]. To take care of these issues, some machine learning calculations are utilized to lead programmed bug triage [1],[3], [4], [5], [6]. The greater part of the bug triage methodologies depend on content classification [3].Be that as it may, these methodologies experience the ill effects of two issues. On one hand, because of the expansive number

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

of bugs, it is important to gather vast scale preparing sets of bugs to acquire great results for bug triage [3]. For instance, in our tests, more than 9000 bug reports and 30000 words are utilized to prepare a classifier for Eclipse. It might cost much time to specifically utilize the extensive scale preparing set in the bug triage process. Then again, the nature of the first bug reports is sufficiently bad. Low quality bug reports might misdirect the triage way to deal with appoint bugs to wrong designers[6].

Essential commitment of this paper is as take after:
Here we are utilizing highlight choice and example determination with verifiable information for decreasing the bug information in bug vault so we get quality information and also low scale information. We are likewise including a chart module for speaking to the bug report's.

- We propose the preparation set diminishment approach for bug triage by consolidating highlight choice with example choice. As far as anyone is concerned, this is the main work to enhance the execution of bug triage by diminishing the size of the preparation set.
- We give a relative study on the impact of various requests of blends. The outcomes demonstrate that the viability of the preparation set decrease can be impacted by changing the request of two.

II. RELATED WORK

J. Anvik, L. Hiew, and G. C. Murphy, [2] creator exhibit a semi-mechanized methodology proposed to rearrange one part of this procedure, the task of reports to a designer for further taking care of. Bug triage intends to apportion a fitting engineer to settle another bug that is to figure out who ought to alter a bug. Creator first proposes the issue of programmed bug triage to diminish the expense of manual bug triage. Displayed methodology depends on a directed machine learning calculation that is connected to data accessible in the bug archive. At the point when another report arrives, the classifier created by the directed machine learning procedure offered a little number of engineers suitable to determine the report.

C. C. Aggarwal and P. Zhao [3], creator presented another worldview for content representation and handling called separation chart representations. Separation diagram representations keep data about the relative requesting and in addition separation among the words in the charts and give a considerably more wealthy representation as far as sentence structure of the gave information. Information revelation from content is conceivable with help of separation chart representation which is unrealistic with the utilization of an unadulterated vector-space representation. Utilization of the separation diagram representation gives huge points of interest from an adequacy viewpoint.\

S. Kim, H. Zhang, R. Wu, and L. Gong [4], creator proposes two plans to manage the clamor present in deformity information. Creator acquainted a system with measure clamor struggle in programming deformity expectation furthermore proposed another technique called CLNI for recognizing uproarious cases in imperfection information. Commotion location and end calculations are proposed to address this issue. Proposed calculation can recognize uproarious information with exactness. Furthermore, in the wake of taking out the clamors utilizing proposed calculation, imperfection forecast precision is moved forward. For the machine learners that don't have solid commotion safe capacity, the clamor disposed of preparing sets created by CLNI can enhance the imperfection forecast execution.

G. Jeong, S. Kim, and T. Zimmermann [5],creator proposed bug hurling chart model can be effortlessly fused into existing bug triaging frameworks. Discover that more than 37 percent of bug reports have been reassigned in manual bug triage to different designers particularly if there should be an occurrence of Mozilla and Eclipse. Proposed model expanded the expectation exactness when contrasted with customary bug triaging approaches. Principle target of proposed strategy is to decrease reassignment in bug triage.

Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis [6], bug hurling is the same as ticket directing that is exchanging an issue ticket among different master gatherings looking for the privilege resolver to the ticket, which is an understood issue in the machine learning writing. Most methodologies use different measurable models to mine work process from

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

movement logs. Creator plan an inquiry calculation, called Variable-arrange Multiple dynamic State look (VMS), that produces ticket exchange suggestions. Creator tended to the likelihood of enhancing ticket directing effectiveness without using so as to get to the ticket content Markov models. In this paper, they exchanged that thought to Eclipse and Mozilla hurling occasions utilizing a changed chart seek calculation. Like their work, proposed approach likewise decreased the length of hurling ways. Moreover, they joined their substance less approach with a substance based way to deal with find a starting designer utilizing a conventional bug task calculation.

C. Sun, D. Lo, S. C. Khoo, and J. Jiang [7], in this paper creator propose a recovery capacity (REP) to distinguish such copies precisely between two bug reports. Proposed methodology is twofold, first BM25F is a powerful literary copies measure that is intended for short unstructured questions and seconds another recovery capacity REP completely using content and other data accessible in reports, for example, item.

Drawbacks of previous system

- Traditional software analysis is not completely suitable for the large-scale and complex data in software repositories.

In traditional software development, new bugs are manually triaged by an expert developer, i.e., a human triager. Due to the large number of daily bugs and the lack of expertise of all the bugs, manual bug triage is expensive in time cost and low in accuracy.

III. BUG TRIAGE SYSTEM

In A time-consuming step of handling software bugs is bug triage, which aims to assign a correct developer to fix a new bug. Here, we address the problem of data reduction for bug triage, i.e., how to reduce the bug data to save the labor cost of developers and improve the quality to facilitate the process of bug triage. Data reduction for bug triage aims to build a small-scale and high-quality set of bug data by removing bug reports and words, which are redundant or non-informative[2,4].

Here, we combine existing techniques of instance selection and feature selection to simultaneously reduce the bug dimension and the word dimension. The reduced bug data contain fewer bug reports and fewer words than the original bug data and provide similar information over the original bug data. We evaluate the reduced bug data according to two criteria: the scale of a data set and the accuracy of bug triage[1]. Here, we propose a predictive model to determine the order of applying instance selection and feature selection. We refer to such determination as prediction for reduction orders. Drawn on the experiences in software metrics, 1 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.

Advantages of Proposed System

- Experimental results show that applying the instance selection technique to the data set can reduce bug reports but the accuracy of bug triage may be decreased.
- The scalability and flexibility will be increased.
- Applying the feature selection technique can reduce words in the bug data and the accuracy will be increased.

IV. EXPERIMENTAL RESULTS

Here, have created feature selection and training set reduction scenarios in two ways for comparing the results with respect to provided input. Various parameters of SVM can be considered in the attempt to optimize the performance of this algorithm. We also varied the € parameter which controls the accepted error of the SVM optimization problem solver. For different values of €, Figure IV.1 (a) shows the dependence of the processing time of SVM on the size of the feature space. Figure IV.1 b) is similar but shows the dependence of the processing time on the number of documents. We have found that € has no influence on maF1 as long as its value was smaller or equal to 0.1. However, when the largest value of € was used, the processing time could be reduced by a factor of four in the best case.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

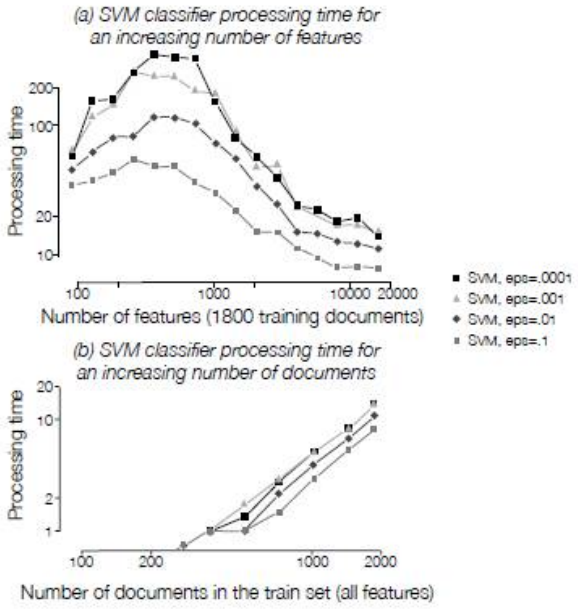

Figure IV.1. SVM processing time (in seconds) for several values of € for an increasing number of features (a) and an increasing number of documents in the training set (b). Experiments were performed on alt.atheism vs. talk.religion.misc from the 20newsgroups dataset.

The above figure shows the processing time of an SVM classifier towards the results. The experiment performed by using the huge dataset of newsgroups.

Sr.No	Category	NB	SVM
1	User Interface Defect	**81.44**	**86.6**
2	Hardware Failure	**84.62**	**95.67**
3	Calculation Defect	**83.57**	**87.11**
4	Database Connection	**82.44**	**83.64**
5	Memory Problems	**81.88**	**84.54**
6	Other	**89.71**	**93.27**

The above table show the detail result of both classifiers scenario with the help of above categories. This result is also calculated with the help of Apache Jmeter. Here the categories are followed as per the type of bugs are resolved. The values comes from data stored related to the category bug records.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

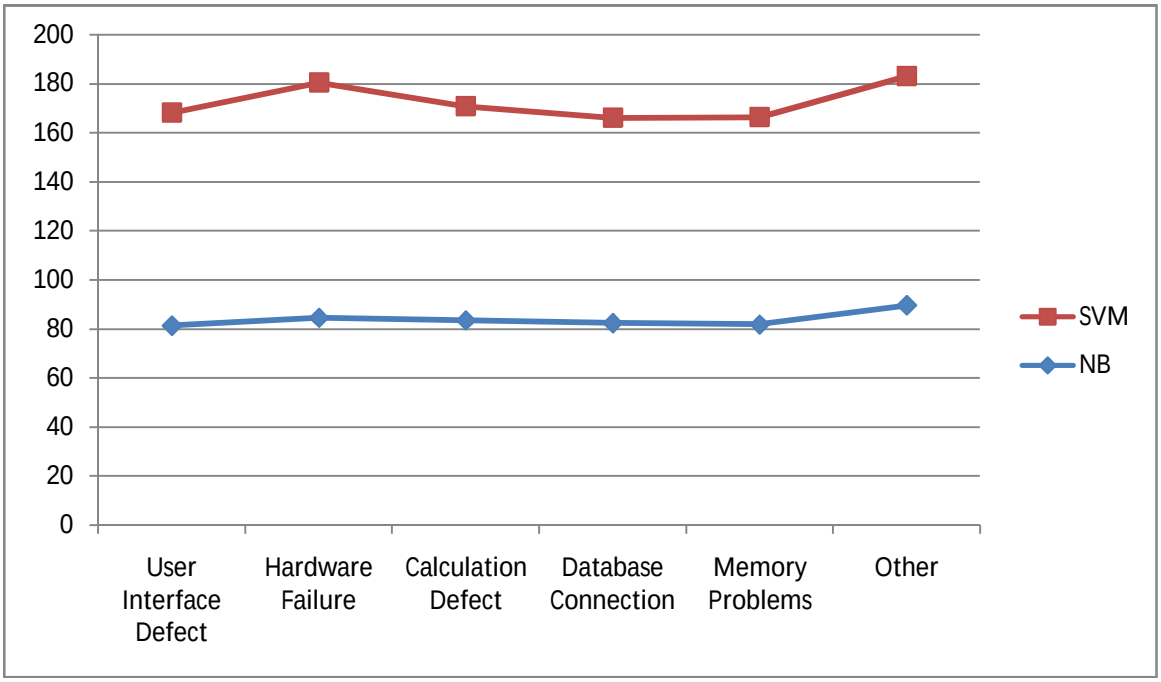

Figure IV.2 Accuracy graph for proposed work

The above Figure IV.2 shows the graph for comparative scenario of classifier Naive Bayes and SVM developed in that is developed in proposed system.

V. CONCLUSION AND FUTURE WORK

Bug triage is an expensive step of software maintenance in both labor cost and time cost. The proposed system aims to form reduced and high-quality bug data in software development and maintenance. Data processing techniques like instance selection and feature selection are used for data reduction. The proposed system can be used for any open source projects that generate huge bug data. Various software companies working on projects like banking, food chain management can use the application of the proposed system. In future we plan on improving the results of data reduction in bug triage to explore how to prepare a high quality bug data set.

REFERENCES

1. Jifeng Xuan, He Jiang, Member, IEEE, Yan Hu, Zhilei Ren, Weiqin Zou, "Towards Effective Bug Triage with Software Data Reduction Techniques", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 1, JANUARY 2015.
2. C. C. Aggarwal and P. Zhao," Towards graphical models for text processing", Knowl. Inform. Syst., vol. 36, no. 1, pp. 121, 2013.
3. J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?",Proc. Intl. Conf. Softwarre Engineering (ICSE 06), ACM, pp. 361-370, 2006.
4. 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. 474494, Jul./Aug. 2010.
5. 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.
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. 11461150, Jun. 2012.
7. D. ubrani and G. C. Murphy, "Automatic bug triage using text categorization", Proc. Intl. Conf. Software Engineering and Knowledge Engineering (SEKE 04), pp. 92-97, 2004.
8. V. Bol on-Canedo, N. S anchez-Maro no, and A. Alonso- Betanzos, "A review of feature selection methods on synthetic data", Knowl. Inform. Syst., vol. 34, no. 3, pp. 483519, 2013.
9. A. K. Farahat, A. Ghodsi, M. S. Kamel, "Efficient greedy feature selection for unsupervised learning", Knowl. Inform. Syst., vol. 35, no. 2, pp. 285310, May 2013.
10. Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning", Knowl. Inform. Syst., vol. 35, no. 2, pp. 249283, 2013.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

11. Q. Hong, S. Kim, S. C. Cheung, and C. Bird, "Understanding a developer social network and its evolution", in Proc. 27th IEEE Int. Conf. Softw. Maintenance, pp. 323332, Sep. 2011.
12. J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques", 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.
13. T. M. Khoshgoftaar, K. Gao, and N. Seliya, "Attribute selection and imbalanced data: Problems in software defect prediction", in Proc. 22nd IEEE Int. Conf. Tools Artif. Intell., pp. 137144, Oct. 2010.
14. G. Lang, Q. Li, and L. Guo, "Discernibility matrix simplification with new attribute dependency functions for incomplete information systems", Knowl. Inform. Syst., vol. 37, no. 3, pp. 611638, 2013.
15. N. Bettenburg, S. Just, A. Schrter, C. Weiss, R.Premraj, and T. Zimmermann, "What makes a good bug report?",Proc. ACM SIGSOFT Symp. Foundations of Software Engineering (FSE 08), ACM, pp. 308-318, Nov. 2008.

BIOGRAPHY

Miss.Varsha T.Bongane received the B.E.(Information Technology) from Savitribai Phule Pune University in 2014 and pursuing M.E. degree in Computer Engineering from Vishwabharati Academy's College of Engineering, Savitribai Phule Pune University,Ahmednagar, Maharashtra, India in 2015-16.

Prof.S.B.Natikar received the M.Tech in computer science and engineering at Basaveshwara Engineering College, Bagalkot under VTU, belgaum. He is working as Assistant Professor in Vishwabharati Academy's College of Engineering, Savitribai Phule Pune University, Ahmednagar, Maharashtra, India. His research interests include networking,wireless network,cloud computing,advanced databases and mobile computing.