ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

Automatic Bug Triage Using Data Reduction Technique

Assit. Prof. Pratima Patil[1], Tejas Shingte[2], Pranav Kamatkar[3], Prachi Tikar[4], Sumit Kulkarni[5]

Assit. Professor, Computer Dept., TAE,, Pune, India [1]

Student, Computer Dept., TAE, Pune, India [2,3,4,5]

Abstract: Bug is a very essential factor which is occur in software. Almost every company facing problem to solve issues related bug, software companies spends over 40-45 percent of cost in dealing with software bugs. Very important step to handle bug is bug triage, which aims to correctly assign a developer to a new bug. To lower the time price in guide work, text category strategies are implemented to behavior computerized bug triage. In this paper, we address the problem of facts reduction for bug triage, i.e., a way to reduce the scale and enhance the satisfactory of bug records. We integrate instance selection with feature selection to simultaneously lessen facts scale on the malicious program dimension and the word measurement. To determine the order of making use of example selection and function choice, we extract attributes from historic malicious program information units and build a predictive model for a new computer virus information set.

Key words: Mining software repositories, application of data preprocessing, data management in bug repositories, bug data reduction, feature selection, instance selection, bug triage, prediction for reduction orders.

I. INTRODUCTION

For handling these software program application repositories, the computer virus repositories play a very critical feature in extraction of the information. Computer virus repositories consist of all bug reports and these computer virus repots are mapped as a file and a particular developer is mapped to the label of the document. Developers similarly to users can put up their defects thru massive open supply task like Mozilla and Firefox due to the truth they comprise large laptop virus repositories to keep all of the computer virus reports. The normal happening error are so huge that it turns into too tough to address the specific issue. The principle objective of the paper is to gain the bug reviews from large informantion units. To get the correct outcomes we're going to get low scale and high exceptional data units with the aid of doing away with the computer virus reviews and phrases which can be redundant and so as to be non-informative. By way of using this method we're going to boom the accuracy of the computer virus reports. We are also set to offer the worm reviews according to a particular area, that's no matter the domain which a enterprise o person may be using for his mission we're going to tackle the results consistent with that specific domain. Bug reports are produced in step with that domain using pinnacle-k pruning set of rules which tackles each report with the help of a rating system.
Bug repository (a standard software program repository, for storing information of insects), plays crucial role in dealing with software program bugs. Software bugs are inevitable and fixing insects is highly-priced in software program improvement. Software groups spend over 45 percentage of value in fixing bugs. massive software initiatives installation bug repositories (also referred to as bug tracking systems) to help facts collection and to help builders to deal with bugs .In a worm repository, a computer virus is maintained as a bug file, which facts the textual description of reproducing the bug and updates according to the repute of bug fixing . A bug repository affords a statistics platform to support many styles of tasks on insects, e.g., fault prediction computer virus localization, and reopened worm analysis. In this paper, malicious program reviews in a worm repository are called computer virus information. There are two demanding situations related to computer virus facts that may affect the powerful use of computer virus repositories in software program development responsibilities, specifically the huge scale and the low great. On one hand, because of the everyday-suggested insects, a large quantity of latest bugs are stored in bug repositories.

Bugs might also deceive related builders at the same time as redundant insects waste the restrained time of malicious program handling. A time-ingesting step of handling software program insects is worm triage, which objectives to assign an accurate developer to restoration a brand new worm. In conventional software improvement, new bugs are manually triaged via an expert developer, i.e., a human cause. Due to the large wide variety of every day insects and the lack of awareness of all the insects, manual malicious program triage is steeply-priced in time value and occasional in accuracy. In guide bug triage in Eclipse, forty four percent of insects are assigned by using mistake at the same time as the time price between starting one computer virus and its first triaging is 93 days on common. To keep away from the pricey cost of guide computer virus triage, current paintings has proposed an automated bug triage approach, which applies textual content category strategies to expect builders for bug reviews. On this technique, a worm file is mapped

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

to a file and associated developer is mapped to the label of the document. Then, malicious program triage is transformed right into a trouble of text type and is mechanically solved with mature textual content type techniques, e.g., Naive Bayes. Based totally at the consequences of text type, a human triager assigns new bugs by means of incorporating his/her knowledge.

II. METHODS AND MATERIAL

The reduction of data can be reduced y using the classifier based on the bug report or bug repository. We can be using some classifier techniques such as the text classification, naïve Bayes classification algorithm. To reduction of data we can use some technique and algorithm. In the bug repositories, all the bug report are filled by the developers in natural languages. To remove the noisy duplicate words and uninformative bug we can use various algorithms Instance selection algorithm, feature selection algorithm. In the instance selection algorithm can be generate the reduced data set based in training set which are generated by the classifier and also feature selection algorithm can be generate reduced data set based on the bug report remove the uninformative bug report. To apply the data reduction to each new bug data set we need "Prediction for reduction order". Given an instance selection algorithm and feature selection algorithm the order can be determine as FS-IS or IS- FS or better one. We simultaneously reduce the word dimension and bug dimension by applying the prediction order

A. Software requirement

- Operating System : Windows Xp/7/8/10
- Technology : Java and J2EE
- Web Technologies : Html, JavaScript, CSS
- IDE : Eclipse Luna
- Web Server : Tomcat
- Database : My SQL
- Java Version : J2SDK1.7

B. Hardware requirement

- Hardware : Pentium Dual Core and Above
- Speed : 2.80 GHz min
- RAM : 1GB min
- Hard Disk : 20 GB min
- Floppy Drive : 1.44 MB min
- Key Board : Standard Windows Keyboard
- Mouse : Two or Three Button Mouse
- Monitor : SVGA

III. PRELIMINARIES AND SYSTEM ARCHITECTURE

The input of the proposed system is in the form of bug data set. The bug data set consists of bug report of large open source project. We also get the all details of the developer who have worked on the respective bug. The bug report is mainly separated in two parts: I. Summary and II. Description. The system is gives predicted results in the form of output as soon as reduced data set.

Module1: Bug Report
In bug report, we considered a bug data set of open source projects such as Eclipse or Mozilla. Bug report contains all types of bug like java, c#. To resolving bug each data set having thousands bug. Bug report is converted into a text matrix with two dimensions namely the bug dimension and the word dimension. Each row of the matrix indicates one bug report while each column of the matrix indicates one word.

Module2: Classification
We are used classification techniques to take the input data set and divide it under different classification label to form the sets. We can do classification that is machine learning such as supervised and unsupervised learning. We can use Naive Bayes to perform text classification.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

Module3: Data Reduction
We are performing data reduction techniques to reduce the scale of bug data sets as well as improve the data quality. We used Instance Selection (IS) algorithm and Feature Selection (FS) algorithm.

Module4: Feature Selection
To remove noisy duplicate words in a data set used feature selection algorithm. By removing uninformative words we can improves the accuracy of bug triage. The algorithms are proceeding as following manner:

- First select a minimum set of features.
- Reduce # of patterns in the patterns which are easy to understand.
- Create new attribute. The attributes capture the important information.
- Use the smallest representation which is enough to solve the task.

Module5: Instance Selection
Instance selection is associated with classification technique. It is a nontrivial process of identifying valid and potentially useful and ultimately understandable patterns in data. It is used to reduce the noisy and redundant instances. It can be provides reduced data.

IV. RESULT AND DISCUSSION

This web based application specially design to minimize workload of developers and efficiently track bugs. Here, three login will be provided for developers, admin, manager. Developer will register his/her profile. After registration , developer will get login credentials. Using login credentials, developer can login to application. List of bugs will be displayed with there bugID. Status of bugs are also mentioned as triage/rectify as shown in fig. 1

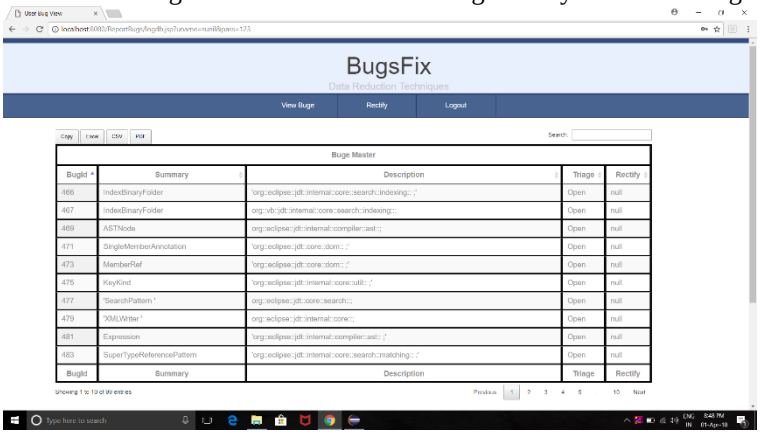

Fig.1

Manger will report bug id, error description and status of bug as shown in fig. 2

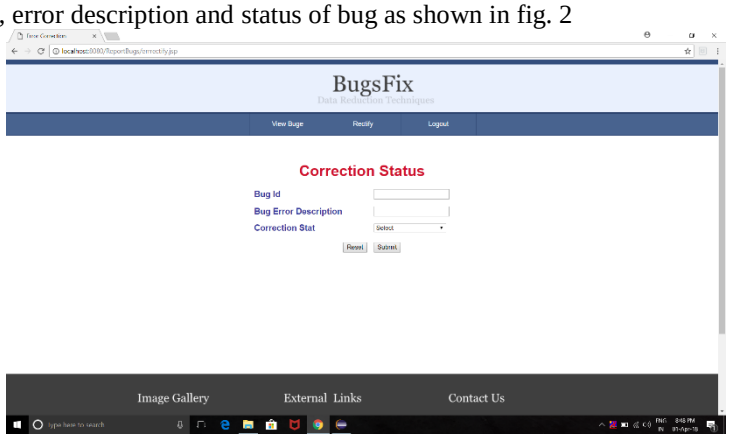

Fig. 2

After this process, important part is data reduction technique. File is needed to browse. Data reduction technique will applied on it. Time slot will be provided.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

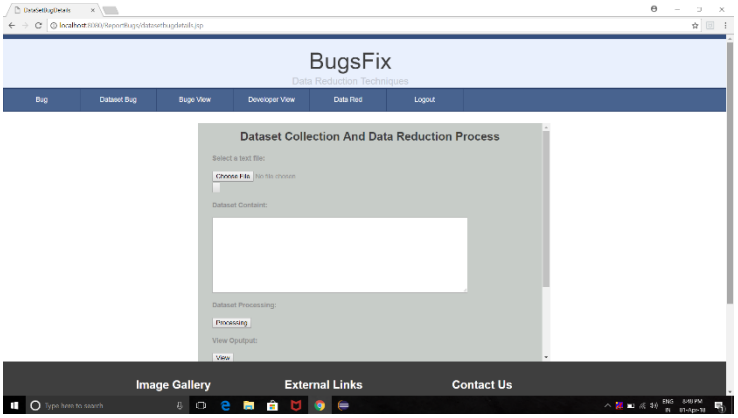

Fig. 3

Result graph will generated. We have consider count vs status. It is shown in fig. 4

Fig. 4

V. CONCLUSION

Computer bug triage is an pricey step of software program upkeep in each labor price and time cost. on this paper, we combine function choice with example selection to lessen the dimensions of worm information sets in addition to enhance the facts high-quality. To decide the order of applying example selection and function selection for a new computer virus records set, we extract attributes of every worm facts set and teach a predictive version based totally on ancient facts sets. We empirically look at the information reduction for trojan horse triage in trojan horse repositories of two big open supply tasks, particularly Eclipse and Mozilla. Our work offers an method to leveraging strategies on facts processing to form reduced and awesome trojan horse statistics in software improvement and preservation. In future paintings, we plan on enhancing the effects of information reduction in computer virus triage to explore the way to prepare a high satisfactory bug records set and tackle a website-precise software program task. For predicting reduction orders, we plan to pay efforts to discover the capacity dating between the attributes of bug records units and the discount orders.

REFERENCES

[1] 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.
[2] 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.
[3] 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.
[4] C. C. Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform. Syst., vol. 36, no. 1, pp. 1–21, 2013.
[5] Bugzilla, (2014). [Online]. Avaialble: http://bugzilla.org/
[6] 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.
[7] 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.
[8] 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.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

[9] S. Breu, R. Premraj, J. Sillito, and T. Zimmermann, "Information needs in bug reports: Improving cooperation between developers and users," in Proc. ACM Conf. Comput. Supported Cooperative Work, Feb. 2010, pp. 301–310.
[10] 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. 483–519, 2013.
[11] 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
[12] 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.
[13] Eclipse. (2014). [Online]. Available: http://eclipse.org/
[14] B. Fitzgerald, "The transformation of open source software," MIS Quart., vol. 30, no. 3, pp. 587–598, Sep. 2006.