Scientific Journal Impact Factor (SJIF) : 3.518
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

ANALYSIS OF AUTOMATIC BUG TRIAGE WITH DATA REDUCTION TECNIQUES

Ramana.R[1] and Krishnamoorthy.S[2]
[1]*PG Student, Dept. of CSE, Bharathiyar College of Engineering and Technology, Karaikal.*
[2] *Professor, Dept. of CSE, Bharathiyar College of Engineering and Technology, Karaikal.*

Abstract−Software world has developed in various things. Still, the bug reports are more difficult job in software sectors. So they used a bug triage method to tackle the process with easy way. In earlier method, bug reports are handled by using a manual triage process. The manual triage process takes more time to complete the process and labor cost also increased. To avoid this step we move on to automatic bug triage with a classifier. Automatic bug triage is the effective step to solve the bug reports which is correctly assigned to the developer for fixing the bug reports and there is an issue of data reduction in bug triage (i.e.) how to decrease the scale and then enhance the bug data. So we use instance selection and feature selection process together. Finally, our work provides a better result with deep learning technique on data processing to form a reduced data with enhanced bug data sets.

Keywords−data mining, bug repositories, bug triage, reduction processing, prediction results

I. INTRODUCTION

Data mining is the efficient methods to solve the software quires. In the software world more problems are happening and difficult to handle. Bug repositories are most useful one to collect the large amount of data or any information related to bug reports and act as a database. Deep learning techniques used in software bug repositories to carry out the software problems. Bugs are usually represents crash, error, failure, freeze, corrupt the file etc.so all the bug reports collected in a bug repositories then it decides whether it old bugs (like historical bug)or new bug and then only we easy to divide the specific developer to each bug reports. Bug triage acts in a speedy way; if the bug reports come from the operator then it takes the problem and works with the trained developer. Finally, we get the accuracy results. Time and cost reducing reason is the more advantages of using automatic bug triage. In the earlier work of bug reports handled by human, so that process is not effective in manner. Human triage process is failure in now-a-days. Bug repositories are used to store the bug reports. Reports contains some mislead data set. That dataset are not used directly to the bug triage. At present we propose an automatic triage with text classifier. Then it shows the expected results.

II. LITERATURE REVIEW

2.1Towards more accurate retrieval of duplicate bug

In this paper, the author proposed a time series classification method. Since it get the simplicity and effectiveness. The efficiency of the classification based on the size of the training set of data dimensionality. The prototype selection and abstraction algorithms are applied on reduced data. Then it produces the result of classification accuracy.

2.2 An approach to detecting duplicate bug reports using natural language and execution information.

In this paper, the author explained the duplicate bug reports weaken the quality of bug data by delaying the cost of handling bugs. To identify the duplicate bug reports, they design a natural language

Processing approach by matching the execution information.

2.3Reducing features to improve code change based bug prediction

In this paper the author purpose reducing features to enhance code change based on bug prediction. In that paper a framework to examine multiple feature selection algorithms and remove noise feature in classification-based defect prediction. It does not contain how to measure the noise resistance in defect prediction and how to defect noise data. Then it shows the better result.

2.4Developer prioritization in bug repositories

In this paper, they proposed a semi-supervised text classification method for bug triage. The problem of labeled bug reports in existing supervised approaches. The new approach of naïve Bayes classifier and that show the maximum result of both labeled and unlabeled bug reports.

2.5 web application using dynamic test generation and explicit-state model checking

In this paper, web page validation is the major process of today's internet. But it cannot handle dynamically generated pages. Then it approaches a dynamic test generation technique for reliable dynamic web applications. The technique used both combined symbolic execution and concrete process for detecting purpose. Finally, the technique generates test automatically and to minimize the constraints.so that the resulting bug reports are reduced and helpful in finding and fixing the faults.

III. PROPOSED METHODOLOGY

The user or operator of handling the software problems is exhausting one and they decide to report the particular error to the bug tracking method. Most of the non-technical people have some trouble, how to solve these problems in software areas. We decide to tackle problem by using automatic bug triage. Our proposed work of automatic bug triage which is correctly assigns a trained developer with respective reports. The purpose of bug triage is used to save the time and cost.

The user or operator sends the bug reports comes under the eclipse(multi-language software development environment),opera(browser),chrome(browser),firefox(browser),etc.and that reports collected in a bug repositories (repositories are acts as a database).But the reports are not give clear idea for the developer so we have added instance and feature algorithm to extract the attributes of historical bug dataset. Both this algorithm helps to reduce the scale and enhance the bug dataset. At last the reports get into text classifier to provide a predicted result of data using deep learning technique.

IV. MODULE DESCRIPTION

4.1Bug dataset

In this module, most of the dataset are collected from open source projects (Mozilla, Eclipse, Opera, and Chrome etc.). Dataset contain noisy words and unnecessary words are recorded in a bug repository. Data set is the initial step of this process to implement the reports.

4.2Bug triage

Triage is useful for collecting enlarge amount of data and then execute the data easily. Assign a correct developer for fixing the bug is called a bug triage. The experienced developer is also used in this process since it easy to analyze the reports and then provide excellent results.

4.3Reduction process

In reduction process, we have to use two algorithms (instance and feature).Instance method- is used to filter the noisy words and useless words. Then it reduces the dimension of vocabulary. Instance selection method (IS) can provide a minimized data set with filtering technique. By using four instance selection methods are iterative case filter, learning vectors quantization, decremented reduction optimization procedure and pattern by ordered projections.

Feature method is used to filter the duplicate words.it is the preprocessing method for choosing a reduced set of features for big dataset. Reduced dataset is representing the feature of original dataset. Then move on to text classification process. Feature selection method (FS) is concentrate on only the analysis of text data. Four feature selection methods are manipulate in text data, namely information gain, x2 statistic; symmetrical uncertainty attribute evaluation and relief attribute selection.

Methods

Instance selection and feature selection
FS→IS = bug data reduction
which first applies FS and then IS.
On the other hand,
IS →FS denotes first applying IS and then FS
In algorithm 1, we briefly present how to reduce the bug data based on FS→IS.
Given a bug dataset, the output of bug data reduction is a new and reduced bug data set.

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 $\mathcal{T}_{FI}$.

Two algorithms are applied simultaneously
Note that step 2, some of bug reports may be blank during feature selection
In work FS → IS and IS→ FS are viewed as two orders of bug data reduction.
To avoid the bias from a single algorithm, we examine the results of four typical algorithms of instance selection and feature selection, respectively.

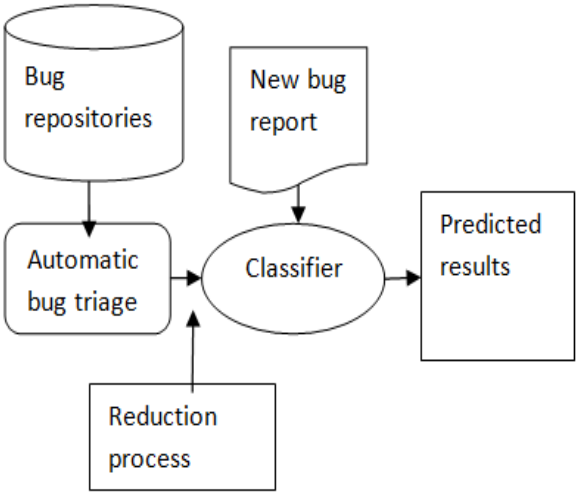

Fig: 1 Architecture system

4.4Classifier

In this module, natural language classifier is used in various sectors like document indexing, unsolicited messages filtering, data maintenance etc. Text classifier is used in the research for bug classification. Finally we added the deep learning then it provides the predicted results.

V. CONCLUSION

Automatic bug triage is the systematic part of software maintenance and it's mainly used for handling the bug reports. Manual triage process does not produce an accurate result. It take too much of time and labor cost also increased. Instance and feature selection methods are also used in this process. Then it reduces the data scale and enhances the data quality. Text classifier to produce the predicted results

REFERENCES

[1] C. Sun, D. Lo, S. C. Khoo, and J. Jiang, "Towards more accurate retrieval of duplicate bug reports," in Proc. 26th IEEE/ACM Int. Conf. Automated Softw. Eng., 2011.
[2] X. Wang, L. Zhang, T. Xie, J. Anvik, and J. Sun, "An approach to detecting duplicate bug reports using natural language and execution information," in Proc. 30th Int. Conf. Softw. Eng., May 2008, pp. 461–470.
[3] S. Shivaji, E. J. Whitehead, Jr., R. Akella, and S. Kim, "Reducing features to improve code change based bug prediction," IEEE Trans. Soft. Eng., vol. 39, no. 4, pp. 552–569, Apr. 2013.
[4] J. Xuan, H. Jiang, Z. Ren, and W. Zou, "Developer prioritization in bug repositories," in Proc. 34th Int. Conf. Softw. Eng., 2012, pp. 25– 35.
[5] 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.
[6] J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?" in Proc. 28th Int. Conf. Soft. Eng., May 2006, pp. 361–370.
[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] C. C.Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform. Syst., vol. 36, no. 1, pp. 1–21, 2013.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] V. Bol_on-Canedo, N. S_anchez-Maro~no, and A. AlonsoBetanzos, "A review of feature selection methods on synthetic data," Knowl. Inform. Syst., vol. 34, no. 3, pp. 483–519, 2013.
[14] 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.
[15] B. Fitzgerald, "The transformation of open source software," MIS Quart., vol. 30, no. 3, pp. 587–598, Sep. 2006.
[16] 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.
[17] N.E.Fenton and S.L.Pfleeger, Software Metrics: A Rigorous and Practical Approach, 2nd ed. Boston, MA, USA: PWS Publishing, 1998.
[18] Y. Freund and R.E.Schapire, "Experiments with a new boosting algorithm," in Proc.13th Int. Conf. Mach. Learn., Jul. 1996, pp. 148– 156.

