A Survey on Automatic Bug Triage Techniques

1Sonali Thaokar, 2Pragati Patil, 3Rahul Pusdekar
1, 2, 3Department of Computer Science & Engineering, Abha College of Engineering, Nagpur, India.

Abstract—The process of fixing bug is bug triage that aims to properly assign a developer to a brand new bug. Software system firms pay most of their value in managing these bugs. To reduce time and price of bug triaging, we present an automatic approach to predict a developer with relevant expertise to resolve the new coming report. In proposed approach we have a tendency to do data reduction of bug data set which can reduce the dimensions of the data still as increase the standard of the data. We are using instance selection and feature selection technique at the same time with historical bug data. We have added a new module here which will describe the status of the bug like whether it assigned to any developer or not and it is rectified or not. In addition, the load between developers based on their experience is re-balanced. The experimental results show that the planned theme will effectively improve the detection performance compared with previous methods.

Index Terms—bug data reduction, feature selection technique, instance selection technique, prediction for reduction orders, bug triage, machine learning techniques.

I. INTRODUCTION

A bug repository plays main position in managing software bugs. Many open source software projects have an open bug repository that makes it possible for each developer and users to publish defects or issues in the software, suggest possible enhancements, and remark on existing bug reports.

In large open source software project have the bug repository that store the details of the bug. For large open source software project, the quantity of every day bugs is so substantial which makes the triaging process more challenging and difficult. There are two challenges associated with bug data that will have an effect on use of bug repositories in software development tasks, specifically the large scale data and low quality data. In a bug repository, a bug is kept up as a bug report, which record in the form of text that reproducing the bug and update as per the status of bug fixing.

In software companies spend their time and cost in dealing with the bug. The process of assigning a potential developer for fixing bug is the bug triage. Handling the software bug by bug triage is time consuming. Manual bug triage is time consuming and low in accuracy. To reduce the cost and increase the accuracy of manually bug triaging in this paper proposed an automatic bug triage.

In this paper an effective bug triage approach is proposed to reduce the bug data to save the labor price of developer and enhance the quality of bug data by eliminating the repetitive and non useful bug reports.

II. LITERATURE SURVEY

In [1] this paper, Software firms spend their time and cost in dealing with software bugs. So that bug triage use for fixing the bug, the goal of bug triage is bug assign the potential developer. To reduce the time and cost in manual work, proposed the automatic bug triage. In this paper used data reduction technique to reduce the data set and improve the quality of data set. Here we combine two reduction techniques, namely instance selection and feature selection. These techniques are used to reduce the data scale on the bug dimension and the word dimension. To focus the request of applying instance selection and feature selection, we extract attributes from historical bug data sets and develop prescient model for new bug data set. The outcome shows that data set can effectively reduce by the data reduction techniques and also increase the accuracy of bug triage. Our work gives approach to leveraging techniques on data processing to form reduced and high-quality bug data in software improvement.

In [2] they mention Open supply development projects most of the time aid an open bug repository to which both developers and users can re-port bugs. The experiences that appear on this repository ought to be triaged to verify if the record is one which requires attention and whether it is, which developer will be assigned the responsibility of resolving the report. Massive open supply developments are pressured by using the rate at which new bug reports appear within the bug repository. In this paper, we present a semi-automated approach intended to ease one part of this method, the assignment of reports to a developer. In this paper used machine learning algorithm that produced the classifier to classify the developers which is potentially solve the report. With this procedure, reach the accuracy level of 57% and 64% on the Eclipse and Firefox development projects. We have now additionally applied our method to the GCC open source development with much less positive outcome. We describe the
stipulations below which the method is applicable and also report on the lessons we learned about applying machine learning to repositories used in open source development.

In [3] proposed the process of fixing the bug is called bug triage which aim to assign the new coming bug to the correct developer. The existing bug triage approach used machine learning algorithms, which construct classifiers from the training sets of bug reports. In observe, these strategies suffer from the large-scale and low-quality training sets. In the proposed work we used both instance selection and feature selection techniques to reduce bug triage. In this paper we studied the combination of feature selection algorithm $\chi^2$-test, instance selection algorithm Iterative Case Filter. We use the eclipse to calculate training set reduction on bug data. For the training set, 70% phrases and 50% bug reports are eliminated after the training set reduction. The experimental results exhibit that the new and small training sets can provide higher accuracy.

In [4] paper, As of late, machine learning classifiers have risen as an approach to anticipate the presence of a bug in a change made to a source code document. The classifier is initially prepared on software history data, and afterward used to foresee bugs. Two drawbacks of existing classifier-based bug expectation are conceivably lacking precision for handy utilize, and utilization of countless. These extensive quantities of components antagonistically affect adaptability and exactness of the methodology. This paper proposes feature selection technique pertinent to classification based bug forecast. This method is connected to foresee bugs in software changes, and execution of Na"ive Bayes and Support Vector Machine (SVM) classifiers is characterized.

In [5] this paper, we propose a semi-supervised text classification approach for bug triage to stay away from the lack of labeled bug reports in existing supervised approaches. This new methodology joins naïve Bayes classifier and desire augmentation to exploit both marked and unlabeled bug reports. This methodology prepares a classifier with a small amount of labeled bug reports. At that point the methodology iteratively labels various unlabeled bug reports and trains a new classifier with marks of all the bug reports. We additionally utilize a weighted proposal rundown to help the execution by forcing the weights of multiple developers in training the classifier. Trial results on bug reports of Eclipse demonstrate that our new methodology outflanks existing supervised approaches in terms of classification exactness.

### III. PROPOSED SYSTEM

The process of settling the bug is the bug triage, which aim to correctly assign a potential developer to a new bug.

Figure 1 shows the architecture of the system. In the proposed system we collect the data set from eclipse. The input contains the bug data. Each bug data has the bug report and every bug report contain the summary and description. Bug reports are unstructured data which may contain irrelevant words. Therefore, we apply the traditional text processing approach to transform the text data into a meaningful representation.

In this propose system there are two users one is developer and other is tester. In this proposed system use the data reduction techniques for decrease the scale of data and improve the accuracy of data. Data reduction techniques are applied to the data preparation of bug triage. Data reduction has two techniques, namely feature selection and instance selection. Both techniques are used for the data processing. The instance selection is used for the bug reports in bug data and the feature selection is used for words in the bug data. In the proposed system both techniques are grouping to use.

Artificial neural network classifier is more accurate as compared to naïve Bayes. As there is no limitation to bug data, testers can add huge number of bugs in the system. This is one of the greatest preferences of the proposed framework. Since the entire bug's data is interested in every one of the developers, it takes less time for the developer to take the choice. Developer can rapidly decide to fix the bug.

The purpose of the bug triage is assigning a potential developer to a new bug. Furthermore, in bug repositories, numerous developers have only fixed very few bugs. Such inactive developers may not provide adequate information for predicting correct developers. In our work we improve the more accuracy of the bug triage by using artificial neural network classifier.

### IV. CONCLUSION

Bug triaging is crucial part in the various software development companies. But the manually bug triaging is very expensive in labor cost and time cost for
software maintenance. So the automatic bug fixing is helpful for the developer to fix the bug.

The proposed system focused on reducing bug data set in order to have less scale of data and quality data for that combine two data reduction techniques, namely instance selection and feature selection techniques. The proposed system can be used any large open source project for hug bug data. The proposed system provides the high quality of data and reduces the scale of data.

REFERENCES


<table>
<thead>
<tr>
<th>S. N.</th>
<th>Title</th>
<th>Year</th>
<th>Techniques used for Bug Triage</th>
<th>Pros</th>
<th>Cons</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Automatic bug triage using text categorization</td>
<td>2004</td>
<td>Naive Bayes Classifier</td>
<td>The bug can be automatically assigned to the potential developer for evaluating all the bug report carefully which saves resources used in bug triage or bug assigning task.</td>
<td>The two problems with this approach is sometimes the developer who fix the bug is not the one to whom it was officially assigned, second the algorithm does not proved to be as efficient as it was thought to be.</td>
<td>In order to improve the approach for handling bug triage is that on should involve the project’s developer too. We should add on factors that would even deal with unlabeled document in corpus.</td>
</tr>
<tr>
<td>2</td>
<td>Who should fixed bug</td>
<td>2006</td>
<td>SVM, Naive Bayes Classifier</td>
<td>A. It has provided help triager to assigning bugs more efficient. B. If company has little knowledge then new triager can work on it.</td>
<td>The processes only work on Eclipse, Firefox, and gcc; it does not work on other projects.</td>
<td>In order to analyzed that the improvement of the bug assignment process and it is found that gcc project is far worse than eclipse and firefox projects.</td>
</tr>
<tr>
<td>3</td>
<td>Modeling Bug Report Quality</td>
<td>2007</td>
<td>Bugzilla</td>
<td>A. It is provide to reducing overall software maintenance cost. B. Usage of this model leads to better Precision and recall result.</td>
<td>This process only work on two projects i.e., Mozilla and Firefox.</td>
<td>The reporter should be provided with Complete guidelines on what all information, attachment they should provide along with the bug report.</td>
</tr>
<tr>
<td>4</td>
<td>Information Needs in Bug Reports: Improving Cooperation</td>
<td>2010</td>
<td>Card sorting technique</td>
<td>In this paper, interaction between developer and user are necessary for fixing the bug in time.</td>
<td>The drawback is that, the result is only applicable for eclipse and Mozilla and might not be other projects.</td>
<td>It is analyzed that, it show the information status of bug report that means if any information is missing to filled by user then it</td>
</tr>
<tr>
<td>Title</td>
<td>Year</td>
<td>Methods</td>
<td>Description</td>
<td>Advantages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------</td>
<td>--------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Between Developers and Users will show a “pending status” and will keep on notifying the reporter about it.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Towards Training Set Reduction for Bug Triage</td>
<td>2011</td>
<td>feature selection algorithm X 2 -test (CHI), instance selection algorithm, Iterative Case Filter (ICF)</td>
<td>Reduce the large scale of training set and decreases the noisy and redundant data in bug triage.</td>
<td>The disadvantage is that the combination is limited for each algorithms. We analyzed that, to reduce training set by using combination of instance selection and feature selection algorithm.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficient Bug Triaging Using Text Mining</td>
<td>2013</td>
<td>Naïve Bayes Classifier, five selection methods are LOR, X2, TFRF, MI, DFS</td>
<td>Automatic assign bug to the potential developer.</td>
<td>Cost increases due to the overloaded work distribution using Naïve Bayes algorithm. It predicts an experienced developer to fix a new reported bug and redistribute the load of overloaded developer. And also analysed x2 selection method is more effective than other for bug assignment.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>An Approach to Detecting Duplicate Bug Reports using N-gram Features and Cluster Chinkage Technique</td>
<td>2014</td>
<td>N-gram Features, Cluster Chinkage Technique</td>
<td>The technique has provided to improve the Detecting Duplicate Bug performance.</td>
<td>This technique is applicable for only AgroUML, Apache and SVN and might not applicable for Eclipse and Firefox. We analyzed here, improvement of classification power for duplication detection by N-gram Features and divergence problem occurs due to N-gram Features is reduced by the Cluster Chinkage.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bug Triaging: Profile Oriented Developer Recommendation</td>
<td>2015</td>
<td>Domain Mapping Matrix (DMM)</td>
<td>Rather than using historical bug report it uses domain mapping matrix for expertise profile of developer maintenance.</td>
<td>The drawback is only chroming bug repository is used. It is analyzed that, To recommend the most suitable developer list for new bug reports.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

◆◆◆