A EFFECTIVE MODEL APPROACH FOR SOFTWARE BUG TRIAGE USING DATA REDUCTION TECHNIQUES

[1]Ashutosh Pawar, [2]Amol Kadam, [3]Shashank Joshi
[1]Student, [2]Associate Professor, [3]Professor
[1]Computer Department,
[1]Bharati Vidyapeeth University, Pune, India

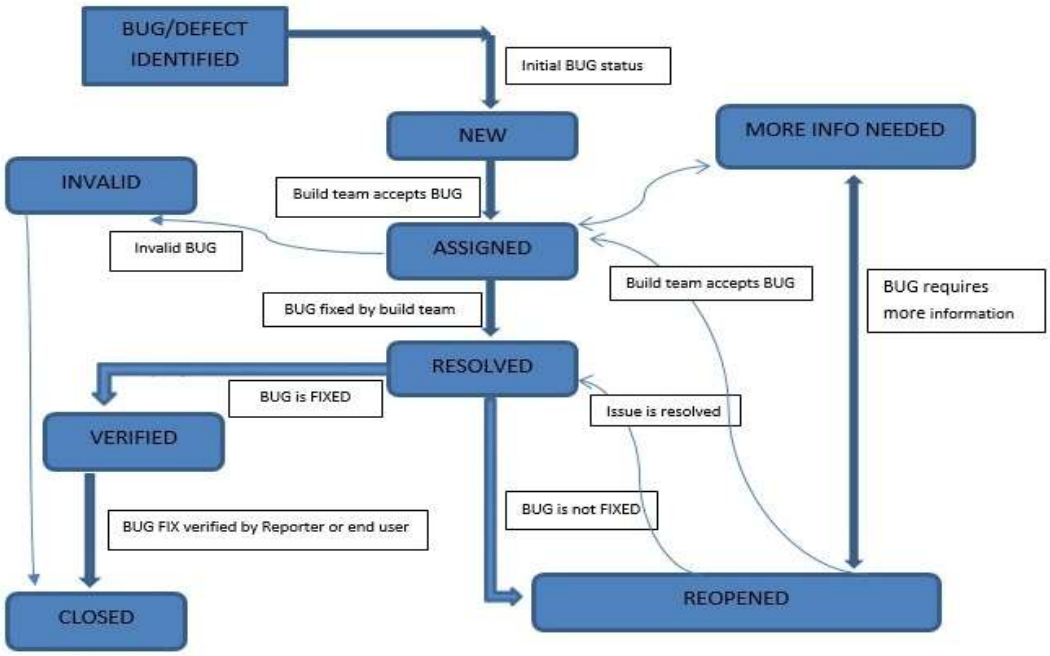
Abstract— Bug analysis and eradication is vital process in software enhancement. Numerous small overlooked bugs give rise to untraceable software failures. Bug triage is an vital phase in software correction. In this phase major objective is to assign a programmer to new defect for future analysis. Today major challenge faced by IT firms is huge cost and time on bug analysis manually. Currently to overcome this challenge two methodologies exists with automated bug analysis namely text classification and binary classification. This article presents model survey on Bug triage domain finding issues and techniques that are currently present and their short coming. Proposed bug triage is handled with data reduction which is outcome of feature extraction and example selection algorithm. Survey has been presented on ten articles and problem definition is been define. The research is been implemented for finite dataset and evaluated on efficiency parameters. Graphical analysis present effectiveness of proposed system.

Index Terms— Data Reduction, Bug Analysis, Bug Removal, Bug Traige, SDLC

I. INTRODUCTION

Today's software development scenario software repositories are huge-scale databanks for stowage of output of software development. Preliminary step in bug warehouse is to accomplish bug reduction. Bug fixative is significant and time-Overwhelming procedure in maintenance of source code. Large development tasks characteristically include an unaddressed bug from huge databank , which together software programmer and end users echo as flaws or issues or bugs or software finding likely improvements and remark on prevailing bug gossips. The benefit of open bug databank is it might permit additional defects to be recognized and cracked, improving e excellence of produced software. Bug triage is greatest vibrant phase for bug fixative, stands to assign a fresh bug to applicable designer for additional treatment. Open source software programmes huge number of defects is produced on large scale which marks triaging procedure very Problematic and perplexing. Major target of bug defect analysis is to allocate a designer for bug analysis and removal. As of a programmer is allocated to a fresh bug she or he would rectify the bug or stab to overcome it. The incentive of effort is to decrease huge gauge of exercise set and to eliminate deafening and terminated bug files for bug defects.

Information reduction is procedure of plummeting bug information by using binary methodologies namely instance assortment and feature assortment that intends to get squat gauge as also excellence data.



BUG LIFE CYCLE Diagram:

II. EASE OF USE

Mining software databanks is interdisciplinary area that aims to hire data mining to handle with Software related problems[3] current software and programmer development large and unseen bug are introduced . Software databanks are used to storage production of source code defects, communications, and requirement specifications. Outmoded software examination is not totally appropriate for large gauge and multifaceted information in software databanks [2]. Information mining has arose as a talented incomes to grip software data. Using data mining methods mining software databanks can expose stimulating evidence in software databanks and resolve actual real software glitches. A issue repository (typical software databank for stowage particulars of defects) theatres an significant part in handling software defects. Program bugs are unavoidable and setting bugs are luxurious in programmer development. Firms devote over 60% of cost in removing bugs [2]. Great software schemes deploy bug databanks (bug monitoring systems) to support material assembly and to support designers to handle defects. In bug databank a defect is upheld in bug report which archives textual account of replicating bug and apprises rendering to position of bug fixative [8]. A bug databank delivers a stage to care numerous kinds of errands on defects, e.g. fault forecast bug localization [2], and revived bug examination called bug data

III. LITERATURE SURVEY

Tabulated survey has been done to find current issue sand problem in bug analysis domain. The issues are top overcome by

Sr.No	Author/Paper	Methodology	Merits	Demerits
1	Dealing with noise in defect prediction	Author introduced a method to measure noise conflict in software defect prediction and also proposed a new method called CLNI for identifying noisy instances in defect data.	Performance and accuracy is improved by using proposed approach	The limitation of their method is that mislabeled instances are often not outliers.
2	Improving bug triage with tossing graphs	Author proposed bug tossing graph model can be easily incorporated into existing bug triaging systems.	Proposed model increased prediction accuracy by up to 23 percentage points Compared to traditional bug triaging approaches. Proposed method is to reduce reassignment in bug triage	Current model is based on regular Markov chains and thus only use the current state for prediction
3	Efficient Ticket Routing by Resolution Sequence Mining	Author design a search algorithm, called Variable order Multiple active State search (VMS), that generates ticket transfer recommendations	Proposed approach is robust to the size, time variability and also reduced the length of tossing paths	Need to extend current model to various mining techniques for better performance
4	Towards more accurate retrieval of duplicate bug reports	Proposed approach is twofold, first BM25F is an effective textual duplicates measure that is designed for short unstructured queries and seconds a new retrieval function REP fully utilizing text and other information available in reports such as product	Improved accuracy of duplicate bug retrieval	Need to speed up the retrieval process.
5	SV kNNC: An algorithm for improving the efficiency of k nearest neighbor	Paper proposed SV kNNC approach for data reduction to enhance performance of kNN	ability to reduce data, classification time required is less and provides best performance because training data are evaluated twice before classification process .	Not all noisy and redundant data is removed by proposed algorithm
6	Finding bugs in web Applications	Dynamic Test generation model	Automatic test generation	Limited tracking in native method of input parameters through database
7	Revisiting bug triage and resolution	Bugzilla approach	Dynamic bug repository	Tedious and longer task.
8	Collaborative bug triage using textual similarities	Collaborative protocol	Providing context for more informed decision	Doesn't provide effective standing

	and change set			point
9	Improving bug traigh through bug tossing graph	Markov model	Prediction increased by 23%	System not representative, programmer retirement information is hidden.
10	COSTRIAE:Bug reporting system	Cost aware triage with CBCF and CF	Reduce sparseness and enhance quality of CBCF	Difficulty in predicting missing values
11	Automatic bug triage using semi-supervised text classification	Navi bayes	Avoid difficulty of labeled reports	Many to one correspondence not possible

Survey Conclusion

Better Data reduction Technique sis been required in bug triage analysis system, dataset needs to be dynamic in nature, constant up gradation. Hybrid approach is required to taken in order to overcome current bug tracking system issue.

Problem Definition

Keep it Simple as you can has been taken up to define problem definition. Proposed research bug tracking system is been designed for input datasets which highly classifies bug and allocates them to required developer and trackes in overall process of bug reduction.

IV. Proposed System

In proposed system, the system proposes the problem of data reduction for bug triage to reduce the scales of data sets and to improve the quality of bug reports. The proposed system uses techniques of instance selection and feature selection to reduce noise and redundancy in bug data sets. The system presents the problem of data reduction for bug triage. This problem aims to augment the data set of bug triage in two aspects, namely to simultaneously reduce the scales of the bug dimension and the word dimension and to improve the accuracy of bug triage. The system proposes a combination approach to addressing the problem of data reduction.

This can be viewed as an application of instance selection and feature selection in bug repositories. The system builds a binary classifier to predict the order of applying instance selection and feature selection. The order of applying instance selection and feature selection has not been investigated in related domains. In this extension, The system add new attributes extracted from bug data sets, prediction for reduction orders, and experiments on four instance selection algorithms, four feature selection algorithms, and their combinations

Advantages:

- It can be used to assist human triages rather than replace them.
- The cost of this prediction is not expensive, compared with trying all the orders for bug data sets.
- It is used to improve the data quality.
- It is used to simultaneously reduce the scales of the bug dimension and the word dimension.

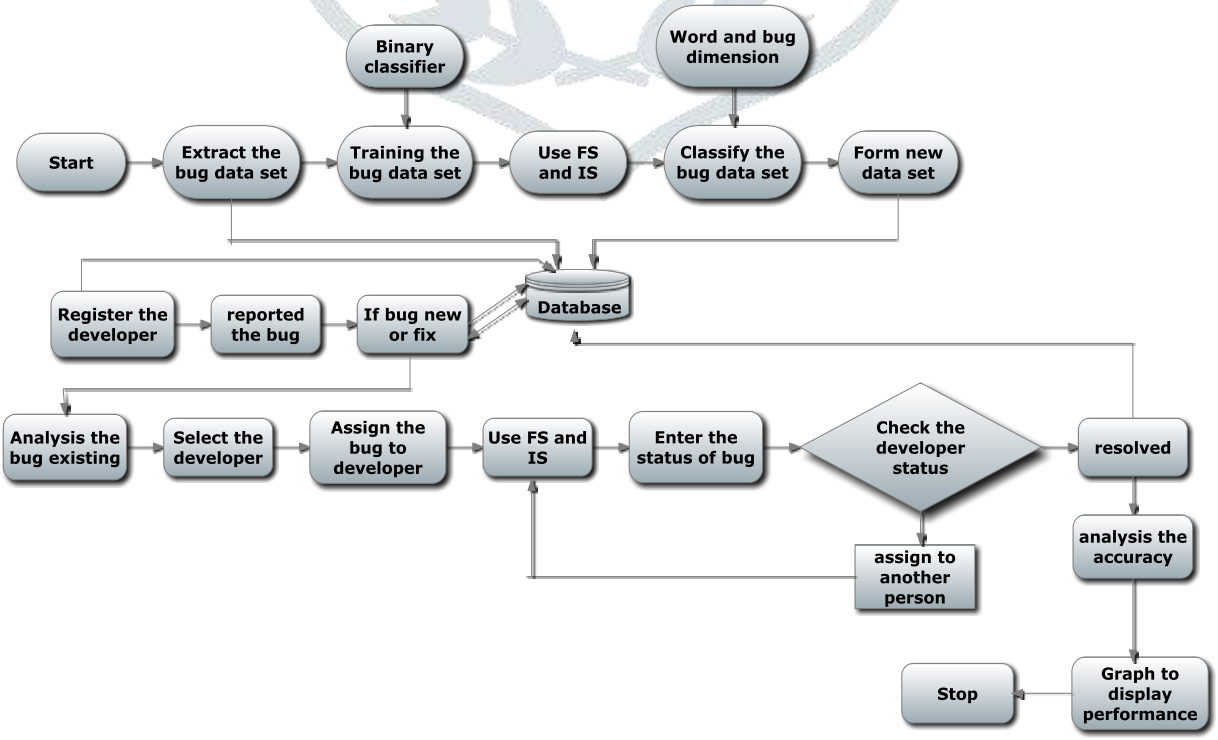


Figure 3: proposed system architecture

V. CONCLUSION AND FUTURE SCOPE

In research system proposed the problem of data reduction for bug triage to reduce the scales of data sets and to improve the quality of bug reports. the system provides an approach to leveraging techniques on data processing to form reduced and high quality bug data in software development and maintenance. The system combine feature selection with instance selection to reduce the scale of bug data sets as well as improve the data quality. Due to the large scales of bug repositories, there exist no adequate labels to mark whether a bug report or a word belongs to noise or redundancy. Since all the bug reports in a bug repository are recorded in natural languages, even noisy and redundant data may contain useful information for bug fixing. To determine the order of applying instance selection and feature selection for a new bug data set, we extract attributes of each bug data set and train a predictive model based on historical data sets. The system construct a predictive model to determine the reduction order for a new bug data set based on historical bug data sets. Attributes in this model are statistic values of bug data sets, e.g., the number of words or the length of bug reports. No representative words of bug data sets are extracted as attributes. The system planned to extract more detailed attributes in future work. This work can be used to assist human triagers rather than replace them

REFERENCES

[1] "Finding bugs in Web applications using dynamic test generation and explicit-State model checking", Adam ki ezun, Julian Dolby, Frank Tip, Danny Dig, 2010.

[2] "Assisting Bug Rep ort Triage through Recommendation", John Karsten Anvik, 2007.

[3] "Towards graphical models for text processing", Charu C. Aggarwal · Peixiang Zhao, 2013.

[4] "Formal Models for Expert Finding in Enterprise Corpora", Krisztian Balog, 2007.

[5] "Hierarchical Language models for Expert Finding in Enterprise Corpora", D Petkova, 2006.

[6] "Overview of Software Fault Prediction using Clustering Approaches and Tree Data Structure", Swati M.Varade, 2Prof.M.D.Ingle, 2012.

[7] "Hyper-Quad-Tree based K-Means Clustering Algorithm for Fault Prediction", Swati Varade, Madhav Ingle, 2013.

[8] "Advances in Instance Selection for Instance-Based Learning Algorithms", Henry Brighton, 2002.

[9] "Information Needs in Bug Reports: Improving Cooperation between Developers and Users", Silvia Breu, Rahul Premraj, 2010.

[10] "A survey on instance selection for active learning", Yifan Fu · Xingquan Zhu · Bin Li, 2013.