Improving Bug Triage Based on Predictive Model

K. Soundarya  
UG Scholar  
Department of Computer Science & Engineering  
Christian College of Engineering & Technology  
Oddanchatram, Dindigul, Tamilnadu-624619, India

M. Soundhariya  
UG Scholar  
Department of Computer Science & Engineering  
Christian College of Engineering & Technology  
Oddanchatram, Dindigul, Tamilnadu-624619, India

V. Vergin Suganya  
UG Scholar  
Department of Computer Science & Engineering  
Christian College of Engineering & Technology  
Oddanchatram, Dindigul, Tamilnadu-624619, India

Mrs. S. Christina Magneta  
Assistant Professor  
Department of Computer Science & Engineering  
Christian College of Engineering & Technology  
Oddanchatram, Dindigul, Tamilnadu-624619, India

Abstract

Lot of MNC companies invest more time and cost to deals with bug report day by day. An necessary step of fixing the bugs is bug triage, which aims to properly assign the programmer for a new bug. To avoid the time increment in manual , text classification and binary classification techniques are used. Data reduction for bug triage is take in hand hence we merge the instance selection and feature selection which concurrently decrease the data size and recover the accuracy of bug reports in bug triage. To find out the order of applying the two algorithms namely CHI feature selection and ICF instance selection by using predictive model to produce high quality of bug data. Bug reports are converted into bar chart model. Finally, produce the output as graphical representation. The output shows that data reduction can successfully reduce the data size and increase the accuracy of bug triage.

Keywords: Bug triage, CHI feature selection, ICF instance selection, data reduction, predictive model

I. INTRODUCTION

Data Mining refer to “Extracting or Mining knowledge from the large database” and also it can defined as the process of analyzing data from different perspectives and summarizing it into useful information. With the help of data mining, one can discover precious information about the customers and their behavior for a specific set of product and evaluate the analyze, store, mine and load data related to them. Many other terms carry a similar or slightly different meaning to data mining, such as knowledge mining from data, knowledge extraction, data/pattern analysis, another popularly used term, Knowledge Discovery from Data, or KDD.

Essential steps in the process of knowledge discovery. Knowledge discovery as a process is depicted in Figure consists of an iterative sequence of the following steps:

1) Data cleaning: to remove noise and inconsistent data
2) Data integration: where multiple data sources may be combined
3) Data selection: where data relevant to the analysis task are retrieved from the database
4) Data transformation: where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance
5) Data mining: an essential process where intelligent methods are applied in order to extract data patterns
6) Pattern evaluation to identify the truly interesting patterns representing knowledge based on some interestingness measures;
7) Knowledge presentation where visualization and knowledge representation techniques are used to present the mined knowledge to the user
A. **Bug Triage**

In Data Mining, Bug is defined as fault occurs in the system over test cause it to fail to meet a sensible user’s expectation of behavior and quality. Data Mining has been used to handle software data. A bug Repository is like software repository which is used for storing details of bugs. In a repository, a bug is maintained as a bug report (A monitored bug is called a bug report to deals with multiple things for detailing the information of reproducing the bug) and bug report in a bug repository are called bug data. Anvik and Murphy [10] explains the detail to reduce the effort of bug triage by making development-oriented recommenders. Furthermore, bug reports in bug triage are converted into documents and bug triage is a type of content-based classification.

II. **RELATED WORKS**

In this part, analysis the previous work on bug report, bug data and bug triage.

[1]. An obtainable approach to semi automating the task of bug report to programmer with the suitable knowledge to resolution the report [2]. propose a method to evaluate noise difference in LNI for determining noisy instances in defect data [3]. creator projected bug tossing graph model can be easily included into existing bug triaging systems. [4]. planned method is healthy to the size, time-inequality and also decrease the length of tossing paths. [5]. Paper proposed SV-kNNC approach for data reduction to enhance performance of kNN. [6]. Improved the accuracy of duplicate bug retrieval. [7]. presented BugMem, a project-specific bug finding tool using memories of bug fixes. BugMem detects potential bugs and suggests corresponding fixes. We evaluate BugMem by computing bug fix memories hit rates. [8]. presented a technique for finding faults in PHP Web applications that is based on combined concrete and symbolic execution. [9]. The analysis performed so far demonstrates that bug report networks are common in the bug report repository. [10]. To help triagers with their work and presents a machine learning approach to create recommenders that help with a variety of decisions motivated at reorganizing the development process.

III. **PROBLEM IDENTIFICATION**

In the existing system,
- Large size and low accuracy, low quality
- Occur time cost and manual cost
- To remove unwanted (noisy) and redundant

In established software development, bugs are manually triaged by an expert developer, i.e., a human triage. It is taken more time and error-flat due to lot of daily bugs is difficult to accurately allot and a human triager is hard to expert the knowledge about all the bugs.

IV. **PROPOSED METHOD**

Here the predictive model is used to find out the order of applying CHI feature selection and ICF instance selection for improving the accuracy of bug report. Data reduction is mainly for decreasing the size of data.

In proposed system, motivation is that noise and redundancy. Because noisy data may deceive the data analysis techniques whereas redundant data may increase the cost of data processing. To handle this situation merge CHI feature selection and ICF instance selection algorithm and produce high quality bug data.

A. **ICF Instance Selection Algorithm**

It is a technique to decrease the number of instance and to improve the quality. ICF is one of the instance selection algorithm related on the predictive model. The methods of ICF consist of two steps, one is noise filtering and instance condensing.
This is diagrammatic representation of bug triage and step by step process of authentication, detailed viewed by Admin, Detailed viewed by Employee and storing the data in database.

**B. CHI Feature Selection Algorithm**

Feature selection is technique used to reduce the attributes of large scale bug report in data preprocessing. CHI is a characteristic feature selection algorithm to measure the relationship between words and programmers.

Algorithm1. Predictive model reduction with CHIFS

\[-\text{ICFIS}\]

- Input: data set $T$ with $n$ words and $m$ bug reports,
- CHI feature selection algorithm $\text{CHIFS}$,
- last number of words $n_F$,
- ICF instance selection algorithm $\text{ICFIS}$,
- last number of bug reports $m_I$,
- Order of the combination $\text{CHIFS} \rightarrow \text{ICFIS}$

**Output:** Predictive model reduction $T_{FI}$ for bug triage dataset

1) Apply FS to $n$ words of $T$ and calculate objective values for all the words;
2) Select the Upper $n_F$ words of $T$ and generate a data set $T_F$
3) Apply IS to $m$ bug reports of $T_F$;
4) Terminate IS when the number of bug reports is equal or less than $m_I$ generate the last data set $T_{FI}$

**V. MODULES**

**A. Password Generation**

To use the username and password for the authentication purpose. The valid users only access the information in the website / webpage. Admin can generate the password for each employee’s based upon their employee id. Every employees have a unique employee id and password.

**B. Task Scheduling**

To provide a user interface to display the project plan and each task state indicating whether the task is inactive or active.

Instance Selection

Instance selection is to obtain a subset of relevant instances (i.e., bug reports in bug data)

**C. Data Reduction**

To reduce the size of data for an input with similar output.

**D. Chart View**

A Project Management application scheduling the report as a table view and chart view for a task state.
VI. RESULT AND DISCUSSION

In this paper, proposed the difficulty of data reduction for bug triage to decrease size of data and increase the quality of bug report. We use the technique of data preprocessing to reduce noise and redundancy in bug data and also predictive model used to find out the order applying the two algorithm already mentioned. The output of the predictive model reduction for bug triage is better than the data reduction technique. Bug triage motivates to assign an suitable programmer to fix a bug report. The reduced bug data consist of less bug reports and words compared with original bug data but given a similar information more than the original bug data.

In upcoming work, prepare to mine more detailed attributes and build predictive model to find out the reduction order for a new-arriving data set. The mining attributes are applied to real world situations.

VII. SIMULATION RESULT

This Fig 3 shows home page of bug report

![Home page](image1)

**Fig. 3: Home page**

This Fig4 provided for authentication purpose. Admin can create a unique password for each programmer during the project registration.

![Login page](image2)

**Fig. 4: Login page**

This process is also done by the admin at the time of registration. Client requirements, product details, product deliver date are observed.

![Project Registration](image3)

**Fig. 5: Project Registration**
Admin can assign the programmer for a particular project.

**VIII. CONCLUSION**

Merging feature selection with instance selection to reduce the scale of bug data sets as well as improve the data quality. Scheduling the report in a table view and chart view for a task, i.e., the output is represented as a Graphical representation so the person who saw the resultant can easily understand.

**REFERENCES**