International Journal of Research
Available at https://edupediapublications.org/journals
Special Issue on Conference Papers
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 06
March 2018

Using Data Mining Techniques in Software Bug Detection

Bhukya. Krishna
Computer Science and Engineering
CMR Technical Campus
Hyderabad, India
krish685@gmail.com

M. Chalapathi Rao
Computer Science and Engineering
CMR Technical Campus
Hyderabad, India.
chalapathiraomarri@gmail.com

***Abstract-*The core objective of software development is to implement high quality software and high quality software is developed using huge amount of software engineering data or information. The software engineering data or information can be used to boost experimentally based generous of software development. The content full information can be fetched using different data mining techniques and functionalities. The Data mining for protected Software Engineering enhances software productivity and quality software engineers are expanding applying data mining algorithms to different software engineering activities. However data mining software engineering data faces numerous challenges, requiring suitable algorithms to dramatically mining the data, reports, documents, graphs and text from such data. Software engineering data introduces code logical data, execution tracking, historical codes and bug data oriented. They accommodate worth of information about a projects condition, advance and expansion. Handling most advanced data mining techniques and thinkers can search the strength of this precious data in order to improve quality of projects to complete in time and range of budget.**

Keywords—Exploratary Data Analysis, Data mining, KDD, Clementine tool, Data mart.

I. INTRODUCTION

A software defect is a bug in a computer environment or system that produces inaccurate or unwanted results, or causes it to react in unintended way. Software bud or defect prediction is the process of finding defective areas in software. It advices to increase software quality and testing capabilities by composing predictive models from code aspects to enable a timely identification of fault-prone modules, it also advises us in planning, monitoring, performing and control the ppredict defect density and to better understand and control the software quality. The Software Defect Prediction outcome, that is the number of defects remaining in a software system, it can be used as an important metric for the software developer, and can be used to control the software process.

a) Clustering

Clustering is a splitting of data into groups of similar objects. Showing the data by some clusters necessarily give up certain important details, but accomplish simplification. It operates data by its clusters. Data modeling keeps clustering in a historical context originated in mathematics, statistics, and numerical data analysis. As a machine learning context clusters associated to hidden patterns, the exploration for clusters is unsupervised learning, and the outcome of the system represents a data models. As an Experimental context clustering plays a wonderful role in data mining applications for example scientific data exploration, information retrieval and data mining, and Web analysis.

b) Classification and Prediction

The main concept is making the data for Classification and Prediction. Making the data involves the consequence activities.

Data Cleaning − Data cleaning involves removing the noise data and missing attribute values. The noise is deleted by applying smoothing techniques and missing attribute values are answered by substituting a missing value with most often occurring value for that attribute. Relevance Analysis − Database may have the redundant data attributes. Correlation analysis is applied to aware whether any two given attributes are related with each other. Correlation is the process used in the real world context but, in data mining, the values of attributes are various types.

II. LITERATURE SURVEY

KDD Process

Primary step in the Knowledge Discovery Process is Data cleaning in data cleaning process noise and missing values are removed. Next process is Data Integration in data integration various data sources are integrates.

Fig: KDD Process

- Data Cleaning-This is the first step of the data processing it is used to remove the noise and inconsistency of the data.
- Data Integration-It is the process used to merge new information to that already exists.
- Data Selection-In this step data relevant to the analysis task are fetched from the data repository.
- Data Transformation-Data is transformed into forms by using aggregate operations
- Data Mining-This is the technique used to extract the knowledge
- Knowledge- In this process knowledge is represented

III. CLASSIFICATION AND PREDICTION ISSUES

- Classification and prediction activities

Data Cleaning − Data cleaning involves eliminating the noise and missing attribute values. The observed noise is eliminated by the smoothing process and the missing attribute values problem is solved by the replacement technique with most often value.

- Relevance Analysis − Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
- Data Transformation and reduction − the data transformed can be done by the following methods.
- Normalization – By using normalization we can transform the data. Normalization involves measuring respective all values for specified attribute. This is opted when in the learning stage involving measurements are used.
- Generalization – In this step the data can be transformed by generalizing concept for this reason the concept hierarchy is used

IV. SOFTWARE BUG COMPLEXTITY ALGORITHM

Defect prevention should begin with an appraisal of the complex risks collaborated with the system. Getting the complex risks stated involves people to aware the possibilities of defects that are often likely to produce and that can have the highest system impact. Plans can be developed to reduce them. If technology can not ensure that defects will not be made, then defects should be identified quickly before the cost-to-fix becomes effective. For the use of this model, a defect has been found when the defect has been generally brought to the presence of the developers, and the developers examine that the defect is valid. A defect has not imperatively been discovered when the user just finds an anomaly with the software. The user has to report the defect and the developers must examine that the defect is valid. A work product is controlled when it reaches a predefined event in its development. This event involves transferring the item from one level of development to the next level. As a work product moves from one event to the next, defects in the work product have a much larger impact on the remaining system, and having changes becomes much more expensive.

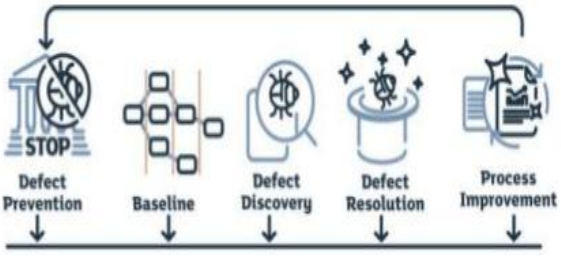

Fig: Bug Detection Process

A work product is subject to change control once it is controlled. This is the activity that is most skipped by companies today, but facilitates one of the greatest areas of payback the study that any defect represents a problem in the process. It is only the developer's fortunate thing that controls a defect from affecting a high level context. Although minimal no of defects represent a chance to learn how to develop the process and reduce large no of failures while the bug or defect in the software is not a big problem means that there are several challenges to the developers to face these kinds of defects from various contextual stages.

V. SOFTWARE DEFECT PREDICTION

A software bug or defect is a abnormal situation in the software environment it to generates incorrect results in most of the cases the flaws are occurred from the programmers mistakes. Software defect prediction is subjected to those flaws in order to predict the failures it consist of independent variables There are different techniques from data mining subject for the defect prediction. Data mining is the analysis process to evaluate the data and it is also used to extract the data patterns from the large data sets. The primary concept of data mining is to discover the knowledge Data mining is further classified into two major tasks such as descriptive and predictive tasks the predictive task is the process to predict the values and the descriptive task is the derived process it has association between the data values. Therefore, defect prediction is very important in the area of software quality and software reliability. Defect prediction is a systematic and scientific research area of software quality engineering. We should concentrate on the following different key points of the problem.\

- Defect Prevention
- Defect Resolution
- Defect Discovering
- Process Improvement

The aim of the defect prevention is to identify the errors and prevent them from occurrence. This will examine the defects and analyze the problem to take necessary actions against to the failures to prevent occurrence of problems it can minimize the rework it will improve the quality of the product some methods and techniques are used to prevent the bugs such as walkthroughs, inspections and root cause analysis etc. Defect resolution process starts when the developer have noticed a valid defect the steps involved in this are prioritized risk, schedule fix etc these are the facilities fixed to enhance the defect resolution process the developer regulates the objective of fixing the defect according to the importance of defect the developer can fix the bug this process is popularly known as schedule fix they schedule when to fix a defect. Defect discovering technology cannot guarantee that defects will not be generated, and then defects should be discovered immediately before the cost-to-fix becomes expensive. User's involvement is a key point in order to find the defect and that is acknowledged by the developer when it is a valid defect as per the study's findings, parties must go back to the process that derived the defect to understand what reason is that behind of a defect. There are various types of errors if we observed those errors we can find more defects and having capability to provide support in the improvement of defect management process lack of knowledge of application, nature of ignorance, lack of information and lack of knowledge of process.

VI. DEFECT DENSITY

Defect Density is the count of defects observed in software development process during a span of time. It enables to conclude when a part of software is ready to be released.

Formula to measure Defect Density

Defect Density = Defect count/ size of the release

Size of release can be calculated in terms of line of code. For example in this analogy we have four modules each module reports different types of bugs with different sizes as prescribed below

M1 = 15 bugs

M2 = 20 bugs

M3 = 30 bugs

M4 = 40 bugs

Total lines of code in each module is shown in below

M1 = 1000 loc

M2 = 1500 loc

M3 = 2000 loc

M4 = 3000 loc

Then, we measure defect density as

Defect count = 15+20+30+40 = 105

Size of the release= 7500 loc

Defect Density= 105/7500= 0.014

The defect density is calculated accordingly as per the values persisted from the data analysis count of defects and size of release is observed to get the final values of the defect density with the specified formula stated above. Defect analysis is a technique which is of key point in Software testing process. The previous defect data is stored for analysis. The identified defects are taken as data and via analysis are done. The data mining implementation is to analyze and predict software bug from huge no of data repository is done to enhance the perfectness and quality of software development.

Acknowledgment

The authors express their sincere gratitude to all those participants who participated in the analysis of their body composition; otherwise without their cooperation this study would not have been possible.

References

[1] Myths and Strategies of Defect Causal Analysis.
[2] The Role of Failure in Successful Design.
[3] Defect Analysis and Prevention for Software Process Quality Improvement",
[4] IEEE Std. 1044-2009: IEEE Standard Classifications for Software Anomalies.
[5] Quality Standards Defect Measurement Manual, United Kingdom.
Marcos Kalinowski, David N. Card and Guilherme H. Travassos, "Evidence-Based Guidelines to Defect Causal Analysis", IEEE Transactions on Software Engineering, 2012.
[6] David N. Card, "Myths and Strategies of Defect Causal Analysis", Proceedings of Pacific Northwest Software Quality Conference, October 2006.
[7] Petrosky.H, "To Engineer is Human; the Role of Failure in Successful Design".
[8] Sakthi Kumaresh and R Baskaran, "Defect Analysis and Prevention for Software Process Quality Improvement", International Journal of Computer Applications, Vol. 8, No.7, October 2010.