Software Bug Detection Algorithm using Data mining Techniques

Safia Yasmeen*
Computer Science Department, OsmaniaUniversity

Abstract— ***The main aim of software development is to develop high quality software and high quality software is developed using enormous amount of software engineering data. The software engineering data can be used to gain empirically based understanding of software development. The meaning full information can be extracted using various data mining techniques. As Data Mining for Secure Software Engineering improves software productivity and quality, software engineers are increasingly applying data mining algorithms to various software engineering tasks. However mining software engineering data poses several challenges, requiring various algorithms to effectively mine sequences, graphs and text from such data. Software engineering data includes code bases, execution traces, historical code changes, mailing lists and bug data bases. They contains a wealth of information about a projects-status, progress and evolution. Using well established data mining techniques, practitioners and researchers can explore the potential of this valuable data in order to better manage their projects and do produce higher-quality software systems that are delivered on time and within budget.***

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

I. INTRODUCTION

A software defect is an error, flaw, mistake, failure, or fault in a computer program or system that produces incorrect or unexpected results, or causes it to behave in unintended way. Software defect prediction is the process of locating defective modules in software. It helps to improve software quality and testing efficiency by constructing predictive models from code attributes to enable a timely identification of fault-prone modules, it also helps us in planning, monitoring and control and predict defect density and to better understand and control the software quality. The Software Defect Prediction result, that is the number of defects remaining in a software system, it can be used as an important measure for the software developer, and can be used to control the software process.

The Data Mining is becoming increasingly popular as a business information management tool where it is expected to reveal knowledge structures that can guide decisions in conditions of limited certainty. Recently, there has been increased interest in developing new analytic techniques specifically designed to address the issues relevant to business Data Mining (e.g., Classification Trees), but Data Mining is still based on the conceptual principles of statistics including the traditional Exploratory Data Analysis (EDA) and modeling and it shares with them both some components of its general approaches and specific techniques [1].

However, an important general difference in the focus and purpose between Data Mining and the traditional Exploratory Data Analysis (EDA) is that Data Mining is more oriented towards applications than the basic nature of the underlying phenomena. In other words, Data Mining is relatively less concerned with identifying the specific relations between the involved variables. For example, uncovering the nature of the underlying functions or the specific types of interactive, multivariate dependencies between variables are not the main goal of Data Mining. Instead, the focus is on producing a solution that can generate useful predictions [1]. Therefore, Data Mining accepts among others a "black box" approach to data exploration or knowledge discovery and uses not only the traditional Exploratory Data Analysis (EDA) techniques, but also such techniques as Neural Network**s** which can generate valid predictions but are not capable of identifying the specific nature of the interrelations between the variables on which the predictions are based.

1.1. Clustering

Clustering is a form of unsupervised learning in which no class labels are provided. It is often the first data mining task applied on a given collection of data. In this, data records need to be grouped based on how similar they are to other records. It is a task of organizing data into groups such that the data objects that are similar to each other are put into same cluster[6][7]. The groups are not predefined. It is a process of partitioning a data in a set of meaningful sub-classes called clusters. Clusters are subsets of objects that are similar. Clustering helps users to understand the natural grouping or structure in a data set. Its schemes are evaluated based on the similarity of objects within each clusters.

1.2. Classification

Classification is a process of finding a set of models that describe and distinguish data classes or concepts. It is the organization of data in given classes known as supervised learning, where the class labels of some training samples are given. These samples are used to supervise the learning of a classification model[8]. Classification approaches normally use a training set where all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model. The model is used to classify new objects. Fraud detection

and credit risk applications are particularly well suited to this type of analysis[8][9]. This approach frequently employs decision tree or neural network-based classification algorithms. The data classification process involves learning and classification.

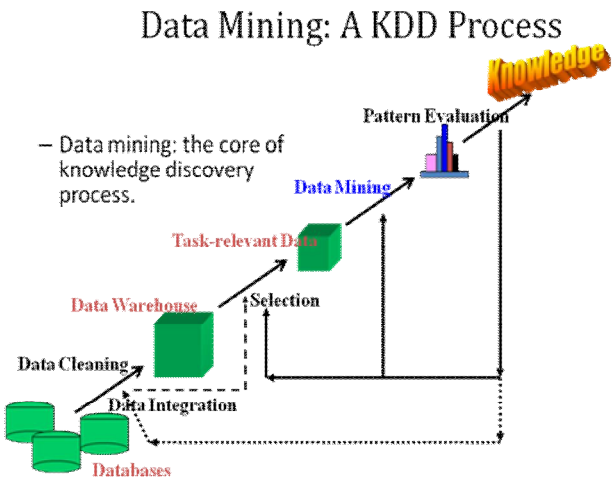
1.3. Association

The Association mining task consists of identifying the frequent itemsets**,** and then forming conditional implication rules among them. It is the task of finding correlations between items in data sets. Association Rule algorithms need to be able to generate rules with confidence values less than one. Association rule mining is undirected or unsupervised data mining over variable-length data and it produces clear, understandable results. The task of association rules mining consists of two steps. The first involves finding the set of all frequent item sets. The second step involves testing and generating all high confidence rules among item sets.

II. LITERATURE SURVEY

As a better tool, recently, many researchers have achieved good results with the aid of chaotic fractal theory as a part of complexity science. The researched results mainly were in the field of graphics and circuit, but some start to explore the issue of software quality by the method of hectic theory [11].

2.1 KDD Process



KDD process

The above figure explains the steps of knowledge data mining process

1. Select functions of data mining(summarization, classification, regression, association, clustering.)
2. Select the mining algorithms.
3. Data mining: search for patterns of interest ,Pattern evaluation and knowledge presentation(visualization, transformation, removing redundant patterns, etc.)
4. Use of discovered knowledge.

2.2 Clementine: A data mining tool

Clementine is a mature data mining toolkit which aims to allow domain experts (normal users) to do their own data mining. IBM SPSS Modeler is a data mining software application from IBM [10]. It is a data mining and text analytics workbench used to build predictive models. It has a visual interface which allows users to leverage statistical and data mining algorithms without programming. SPSS Modeler has been used in these and other industries:

- Customer relationship management (CRM)
- Fraud detection and prevention
- Optimizing insurance claims
- Risk management
- Manufacturing quality improvement
- Healthcare quality improvement
- Forecasting demand
- Law enforcement and border security
- Education
- Telecommunications

SPSS Modeler was originally named SPSS Clementine by SPSS Inc., after which it was renamed PASW Modeler in 2009 by SPSS.[8]

It was since acquired by IBM in its 2009 acquisition of SPSS Inc. and was subsequently renamed IBM SPSS Modeler, its current name. It has a visual programming or data flow interface, which simplifies the data mining process, Clementine is a data mining workbench that enables user to quickly develop predictive models using business expertise and deploy them into business operations to improve decision making. Designed around the industry-standard CRISP-DM model, supporting the entire data mining process, from data to better business results [10]

III. SOFTWARE BUG DETECTION ALGORITHM

Software bug detection algorithm can be carried out in three steps

1. Data collection.
2. Data validation.
3 .Report and Feedback.

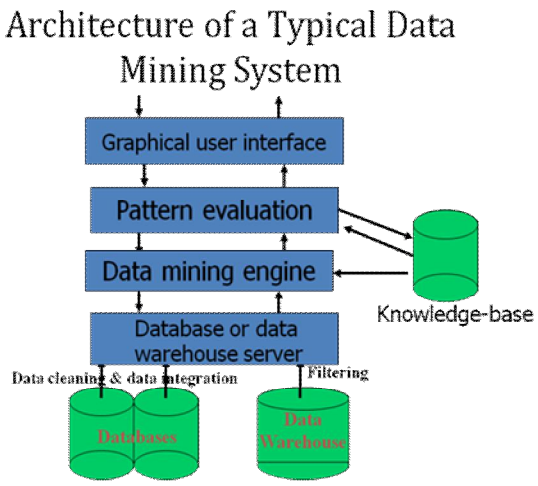


Fig 1 : Architecture of the data mining techniques

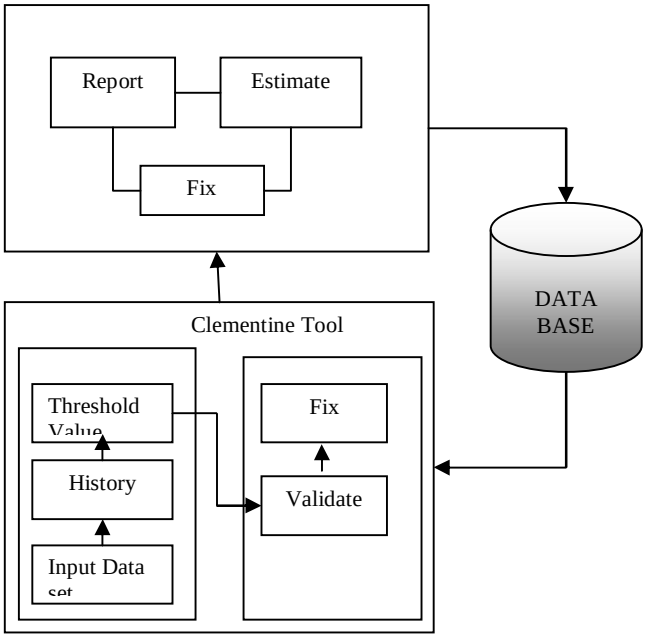


Fig 2: Software bug detection algorithm

In fig 1 and figure 2: depicts the software bug detection using the Clementine tool the figure explains how the procedure is carried out using the data mining tool.

3.1 Data collection mode
Data is captured periodically from the customer r Online Transaction Processing (OLTP) systems providing defect and growth information.

Data is captured regularly from time to time and piled up indefinitely in a data mart. While the information in the OLTP systems continues to change from moment to moment, periodically based on a particular interval the data is recorded in data base Then the data which is stored is supplemented in the Clementine data miningc tool which takes in the various input field like defects, errors arouse in the system, sequence of bugs over a time period, user inputs, etc. Once the data is collected and stored in database the training of the predictive model is done various input data fields are selected and the threshold value is set accordingly for the induction model in Clementine tool for modeling.The models are generated accordingly by the tools after the training phase.

3.2. Data validation
The model generated from the training phase is mapped with the historical data and the defects are identified. The data validation models are generated again and again and matched with the defects which are not seen and then reported to the developer for assessment. The data validation is done regularly and studied against the threshold value over a time interval.

3.3 Report generation:
The reports are generated and evaluated base on the historical dataset with minimum risk scale vs. number of defects. Then the defects are fixed with a feedback to the tool to repeat the test for improving the software defect prediction rate. The statistical study can also be carried on based on defect tracking w.r.t the minimum risk factor and bug reduction and reliability of software can be studied over a period.

IV. CONCLUSIONS

The above algorithm is basic approach for bug detection first, finding as many related defects as possible to the detected defect(s) and consequently makemore effective corrections to the software. This may be useful as it permits more directed testing and more effective use of limited testing resources. Second, helping evaluate reviewers' results during an inspection. Thus, a recommendation might be that his/her work should be reinspected for completeness. Third, assisting the managers in improving the software process through analysis of the reasons why some defects frequently occur together. If the analysis leads to the identification of a process problem, managers can devise corrective action. In future study the algorithm can be enhance more logically by using various mapping technique.

V. REFERENCES:

[1]. Data Mining Techniques for Software Defect Prediction, Ms. Puneet Jai Kaur1, and Ms. Pallavi 2. IJSWS
[2].Tao Xie, Jain Pei, Ahmed E Hassan, "Mining Software Engineering Data", IEEE 29th International Conference on Software Engineering ICSE 07.
[3]. Francisco P.Romero, Jose A.Olivas, MArcele Genero, Mario Piattini, "Automatic Extraction of the main terminology used in Empirical Software Engineering through Text Mining Techniques" ACM ESEM 08 pp. 357 – 358.
[4]. P.Huang and J.Zhu,"Predicting Defect-Prone Software Modules at Different Logical Levels", International Conference on Research Challenges in Computer Science, 2009. ICRCCS '09, pp.37 - 40.
[5]. S.Shivaji,E.J. Whitehead,R.Akella and S.Kim, "Reducing Features to Improve Bug Prediction", 24th IEEE/ACM International Conference on Automated Software Engineering, ASE'09, pp.600- 604.
[6]. A.K. Jain and R.C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, N.J.: Prentice Hall, 1988.
[7]. A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," ACM Computing Surveys, vol. 31, no. 3, 1999.
[8]. J. McQueen, "Some Methods for Classification and Analysis of Multivariate Observations," Proc. Fifth Berkeley Symp. Math. Statistics and Probability, pp. 281-297, 1967.
[9]. H. Spath, Cluster Analysis Algorithms for Data Reduction and Classification of Objects. Chichester: Ellis Horwood, 1980.
[10] /www.ibm.com/software/analytics/spss/A/modeler/-- by IBM.
[11] Zhou Feng zhong, Li Chuan-Xian, A Chaotic Model for Software Reliability, Chinese Journal of Computers, 24(3),(2001), 281-291(in Chinese).

