A Novel Approach for Identifying Software Fault Prediction in mining

Menka Gupta  
Department of Computer Science  
AISECT UNIVERSITY  
Bhopal, India

Dr. Pratima Gautam  
Department of Computer Science  
AISECT UNIVERSITY  
Bhopal, India

Abstract—Identifying and locating defects in software projects is a difficult work. In particular, when project sizes grow, this task becomes expensive. The aim of this research is to establish a method for identifying software defects using data mining applications methods. In this work we used Synthetic data Program (SD). We used mining methods to construct a two step model that predicts potentially defected modules within a given set of software modules with respect to their metric data. The data set used in the experiments is organized in two forms for learning and predicting purposes; the training set and the testing set. The experiments show that the two step model enhances defect prediction performance.

Keywords- Fault Prediction, Hardware, Software, Mining, Fault Detection.

I. INTRODUCTION

Software faults can be classified into two categories. Faults can be classified according to their phase of creation or occurrence, system boundaries (internal or external), domain (hardware or software), phenomenological cause, intent, and persistence. The discussion below is focused on software fault classification based on their recovery strategies. Software fault tolerance techniques are designed to allow a system to tolerate software faults that remain in the system after its development. Software fault tolerance techniques are employed during the procurement, or development, of the software. When a fault occurs, these techniques provide mechanisms to the software system to prevent system failure from occurring. Software permeates our modern society.

Fault prediction is a complex area of research and the subject of many previous studies. Software practitioners and researchers have explored various ways of predicting where faults are likely to occur in software to varying degrees of success. These studies typically produce fault prediction models that allow software engineers to focus development activities on fault prone code, thereby improving software quality and making better use of resources.

II. MOTIVATION

Software metrics and fault data belonging to a previous software version are used to build the software fault prediction model for the next release of the software. Until now, different classification algorithms have been used to build this kind of models. However, there are cases when previous fault data are not present; and hence, supervised learning approaches cannot be applied. In this study, we propose a Feature selection method. In addition we apply a neural gas own as predictive clustering in which firstly we identify the improved cluster and then the mean value of cluster against the threshold value. Then we will apply linear regression for finding the prediction we apply in mat lab. We apply this entire algorithm in Synthetic dataset.

III. RELATED WORK

Olli Simula [2] they proposed The Self-Organizing Map (SOM) for analysis and visualization of high-dimensional data. In this paper, SOM based methods are applied in analysis, monitoring and modeling of complex systems [2]. The SOM provides data-driven approach to process monitoring. When using the SOM, it is not necessary to done the process model analytically. The SOM has the desirable feature of describing nonlinear relationships between a large number of parameters and variables of complex systems phenomenological. By using a history of measurements, dynamical behavior of the process can be introduced into the map, or a set of maps. This approach has been used to model the sequence of states and based on that to predict the future state in the system operation. The SOM facilitates visual understanding of processes. TUBITAK [3] they proposed on the high-performance fault predictors based on machine learning such as Random Forests and the algorithms based on a new computational intelligence approach called Artificial Immune Systems [3]. We used public NASA datasets from the PROMISE repository to make our predictive models repeatable, refutable, and verifiable. The research questions were based on the effects of dataset size, metrics set, and feature selection techniques. In order to answer these questions, there were defined seven test groups. Additionally, nine classifiers were examined for each of the five public NASA datasets. Mahmud [4] has proposed the capability of SVM in predicting defect-prone software modules and compares its prediction performance against eight statistical and machine learning models in the context of four NASA datasets[4]. The results indicate that the prediction performance of SVM is generally better than, or at least, is competitive against the compared models. Kaur, D [5] they have proposed K-Sorensen-means clustering that uses Sorensen distance for
calculating cluster distance to predict faults in software projects. Proposed algorithm is then trained and tested using three datasets namely, JM1, PCI [5] and CM1 collected from NASA MDP. From these three datasets requirement metrics, static code metrics and alliance metrics (combining both requirement metrics and static code metrics) have been built and then K-Sorensen-means applied on all datasets to predict results. Alliance metric model is found to be the best prediction model among three models.

IV. PROBLEM STATEMENT

Two types of research can be studied on the code based metrics in terms of defect prediction. The first one is predicting whether a given code segment is defected or not. The second one is predicting the percentage of the possible defect. Estimating the defect in a given software project provide reliability of the project. Our work in this research is primarily focused on the second type of predictions.

The data sets reported: It is not always clearly reported in papers whether the data analyzed is from new or evolving systems. The basis of data sets has important implications for the type of metrics that should be used, as change data is highly relevant to evolving system though not relevant to new systems. This lack of information reported in reviewed studies limits the conclusions that we can draw from such studies. Also most studies assume the severity of faults to be equal, yet clearly some faults will be more severe than others in terms of how they affect the running of the system and how long they take to fix.

V. PERFORMANCE MEASURES

Measuring the predictive performance of models is an essential part of model development and subject to on-going debate in the literature. We base the following overview of performance measures on work. Reporting performance is often based on the analysis of data in a confusion matrix as shown in Table I and explained further in Table 2. This matrix reports how the model classified the different fault categories compared to their actual classification (predicted versus observed). Many performance measures are related to components of the confusion matrix shown in Table 2. These components can be either within a confusion matrix or used individually. Confusion matrix measures of performance are most relevant to fault prediction models producing categorical outputs, though continuous outputs can be converted to categorical outputs and analyses in terms of a confusion matrix.

<table>
<thead>
<tr>
<th>Construct</th>
<th>Also Known as</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>False Positive</td>
<td>FP and Type</td>
<td>Classifies non Faulty unit as faulty</td>
</tr>
<tr>
<td>False Negative</td>
<td>FN and type ii error</td>
<td>Classifies faulty unit as not faulty</td>
</tr>
<tr>
<td>True Positive</td>
<td>TP</td>
<td>Correct classified as faulty</td>
</tr>
<tr>
<td>True Negative</td>
<td>TN</td>
<td>Correctly classified as non faulty</td>
</tr>
</tbody>
</table>

VI. PROPOSED WORK

The aim of this research is to establish a method for identifying software defects using data mining applications methods such as feature selection method. In this work we used Synthetic data Program (SD). We used mining methods to construct a two step model that predicts potentially defected modules by cross validation method within a given set of software modules with respect to their metric data. The data set used in the experiments is organized in two forms for learning and predicting purposes; the training set and the testing set. The experiments show that the two step model enhances defect prediction performance.

![Figure 1: Shows the tree form](image-url)
VII. CONCLUSION

In this Paper, we try to propose a new defect prediction model based on linear regression methods. LRM and decision tree results have much more wrong defect predictions when applied to the entire data set containing both defected and non-defected items. The data set is already 50% non-defected. Some machine learning method does not try to lean or predict more fault when we put defected or non defected dataset. On that time they fail in Prediction. They sometime did not provide accurate prediction. Therefore the learning methodology fails. So we use Different methodology which can manage such data set for software metrics is required to predict more accurately.

VIII. REFERENCES

[1] www.google.com
[4] Predicting defect-prone software modules using support vector machines Mahmoud O. Elish, Information and Computer Science Department, King Fahd University of Petroleum[2008]