



Improved Default Prediction in Software Module by using Feature Extraction and Adaptive-Boost Learning Approach

Punika Mahajan

M.Tech in Computer Engineering, University College of Engineering,
Punjabi University, Punjab, India

Abstract - An important goal during the cycle of software development is to find and fix existing defects as early as possible. This has much to do with software defects prediction and management. Nowadays, many big software development companies have their own development repository, which typically includes a version control system and a bug tracking system. There is no doubt that these systems proved useful for software defects prediction. The complexities of software development bring a lot of issues which are related with software defects. We have to consider these issues as much as possible to get precise prediction results, which makes the modeling more complex. In this synopsis, challenges are shown in software module default prediction and overcome by our approach (Adaptive Boost with SVM-RBF Kernel). In proposed Methodology a way to predict software defects classification based on mining software repository. A way to collect all the defects during the development of software from the WEKA and MATLAB.

Keywords - Adaptive Boost, Data Mining, Feature Extraction, Machine Learning, Principle Component Analysis (PCA), Support Vector Machine (SVM).

I. INTRODUCTION

It is valuable to predict the software that is defect-prone. There have been many studies and learning approaches that are used to measure the performance of software. Analysis of all required features of defect prediction are used to determine that what factors influence predictive performance. The quality of the software can be measured with the different features such as cyclomatic complexity, design complexity, effort, time estimator, length of the program, operands, operators, line count etc.

1.1 Software Engineering: Introduction

Software Engineering is defined as the systematic and well defined approach to the development, operation, maintenance and retirement of the software. By the word 'systematic' means that the methodologies used for the development of the software are repeatable. The goal of software engineering is to take software development closer to science and engineering that solves the problems of the clients and away from those approaches for development whose outcomes are not predictable.

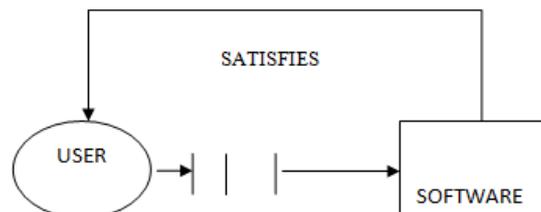


Fig : Software Engineering

1.1.1 Software Quality Attributes

Software Quality attributes can be defined as follows :

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

1. **Functionality** : It is the capability that provides functions which meets the defined and implied needs of the software when it is used.

2. **Reliability** : It is the capability that maintains the specified level of performance.

3. **Usability** :The capability to be understood , learned and used.
4. **Efficiency** : It is the capability to measure the performance relative to the amount of resources used.
5. **Maintainability** : It is the capability to be updated and modified for purposes of making corrections , improvements or adaptation.
6. **Portability** : It is the capability to be adapted for different environments without applying actions.

1.2 Defect Prediction in Software module

a) Data cleansing process

- Initial Preprocessing of the data
- Removal of Constant attributes
- Removal of Repeated attributes
- Replacement of Missing Values
- Enforce Integrity with Domain Specific Expertise
- Removal of Repeated and Inconsistent Instances

b) Prediction Performance measures

- Precision
- Recall
- Accuracy

c) Data Extraction

- Classifier family
- Data set family
- Metric family
- Researcher Group

1.3 Machine learning

Machine learning is a science that explores the building and study of algorithms that can learn from the data. Machine learning process is the union of statistics and artificial intelligence and is closely related to computational statistics. Machine learning takes decisions based on the qualities of the studied data using statistics and adding more advanced artificial intelligence heuristics and algorithms to achieve its goals. Machine learning tasks are classified into three broad categories :

- Supervised learning.
- Unsupervised learning
- Reinforcement learning

Supervised learning : Dataset is presented with example inputs and their desired and defined outputs.

Unsupervised learning : No labels are given to the learning algorithm. Data sets are assigned to segments.

Reinforcement learning : Dataset interacts with a dynamic environment in which it must perform a certain goal without any label.

1.4 Data mining

Data mining is related with the discovery of new and interesting patterns from large data sets for analysis and executive decision making. Data mining is described as the union of past and current or recent developments in statistics, artificial intelligence and machine learning.

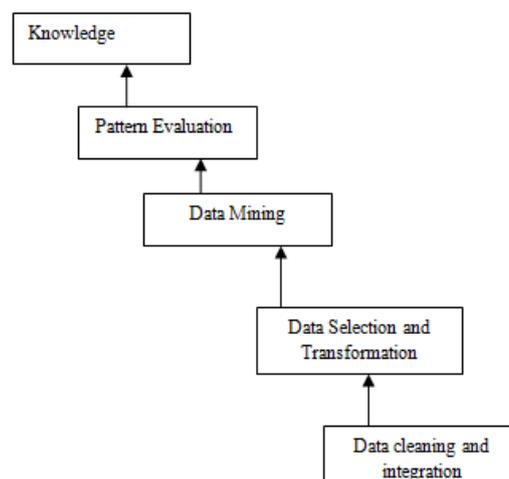


Fig :Steps for Extracting Knowledge from Data

1.4.1 Scope of Data Mining

- Automation in prediction of behavior and trends.
- Automated discovery of previously unknown patterns.

Automation in prediction of behavior and trends

This is the process of finding targeted information in large databases. Predictive problems include forecasting, insurance analysis for prediction and decision making, income tax department of government for fraud discovery.

Automated discovery of previously unknown patterns

This is the process of identifying previously hidden patterns in first step. Pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors.

1.4.2 Data Mining Process

Data mining consists of five major elements:

- Extraction and transformation of data onto the data warehouse system.
- Run data on multidimensional database system in a managed way
- Providing data access to business analysts and other professionals
- Data analyzing
- Presentation of data in useful and required formats such as tables and graphs.

1.4.3 Goals of Data Mining

- Prediction
- Identification
- Optimization
- Classification

Prediction : How certain attributes within the data will behave in the future.

Identification -- Use data pattern to identify the existence of an item, an event, or an activity

Optimization – Optimize the use of limited resources such as time, space, money, or materials and maximize output variables such as sales or profits under certain constraints.

Classification -- Partition data so that different classes or categories can be identified based on combinations of parameters.

1.4.4 Classification Algorithms

- Statistical Algorithms
- Neural Networks
- Genetic algorithm
- Decision trees
- Nearest neighbor method
- Rule induction

Statistical Algorithms :Systems like SAS and SPSS have been used by analysts to detect unusual hidden patterns and explain patterns using statistical models such as linear models.

Neural Networks : Artificial neural networks have the pattern-finding capacity. Neural Network algorithms can be applied to pattern-mapping. Neural networks works successfully on the applications that involves classification.

Genetic algorithms : Genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.

Decision trees : Tree-shaped structure represent sets of decisions . These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID).

Nearest neighbor method : A technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset. Sometimes called the k-nearest neighbor technique.

Rule induction : The extraction of useful *if-then* rules from data based on statistical significance.

II. LITERATURE REVIEW

Many researchers have worked on the defect prediction of software module by using various methods and learning approaches . In this report, main focus is to reduce the complexity of processing by feature extraction method and boundary condition problem resolved as Support Vector Machine (SVM) and component learning by using adaptive boost with SVM -RBF(Radial Basis Function) Kernel. Their work is reviewed and defined as under:

David Gray et al. (2011) have explained the reason of significant preprocessing of data set for suitability of defect prediction. Researchers need to analyze the data that how it will be used by removal of constant attributes, repeated

attributes, missing values and inconsistent instances. The experiments that have been used are based upon NASA metrics data program that results in errors findings and conclude that errors are mainly because of repeated data points[2].

Tim Menzies et al. (2011) describes that how to improve the effort estimates of a project and defect predictions of a software module. The best thing can do to control cost and defects is to discard the needless functionality by making the lines of code to minimum. It has been seen that local treatments are always superior and different to the global treatments because data that appears to be useful in global context is often irrelevant to the local contexts[3].

Qinbao Song et al. (2011) describes the framework that comprises scheme evaluation and defect prediction components. Analyzing the prediction performance for the given historical data sets is done by scheme evaluation and defect predictor constructs models according to the evaluated learning scheme and predicts defects of the software with new data according to the defined constructed model. It has been shown that different learning schemes should be used for different data sets[4].

Ming Li et al. (2012) states that software quality can be controlled by software defect prediction. The defect prediction techniques used currently are based on large amount of historical data but in case of new projects and for many organizations historical data is often not available. In that case, sample based methods for defect prediction can be used by selecting and testing a small percentage of module and after that build a defect prediction model to predict defect proneness of the other modules[7].

Martin Shepperd et al. (2014) have discussed about the factors having largest effect on the predictive performance of the software by conducting a meta analysis of all relevant and high quality primary studies of defect prediction on software module. The experimental results showed that the major factor is the researcher group instead of choice of classifier on the software performance[1].

G Czibula et al. (2014) focuses on the problem of importance during the time software evolution and maintenance for the problem of defect prediction. The quality of the software system is improved by identifying the defective software modules. Relational association rules are the extension of ordinal association rules that are being used in the prediction of software module that whether it is defective or not. It describes numerical range between the attributes that are commonly occur over a dataset by the experimental evaluation on the NASA datasets as well as a comparison is performed to similar existing approaches is provided[5].

A Okutan et al. (2014) states different software metrics that are used for defect prediction and defines the set of metrics that are most important for predicting the defectiveness in the software module. The two more metrics i.e. number of developers and the source code quality are defined other than the promise data set. Experiments results that lines of code and lack of coding quality are the most effective metrics whereas coupling between objects and lack of cohesion of methods are less effective metrics on defect proneness[6].

III. ALGORITHM

Input : Labeled Data Set of Software Prediction

Output : Predict Label of featurer

Software_Module_AdaptiveBoost(Training Set, Test Set)

Begin

for (I=1; I<Len (Training Set); I++)

Begin

Input in Adaptive with SVM Classifier

End

Model of SVM-ADA Classifier

for (I=1; I<Len (Test Set); I++)

Begin

Test on Model of SVM-ADA Classifier

End

Output the Label of Test Set

Analysis the Label by Precisiom, Recall, Accuracy

End

IV. METHODOLOGY

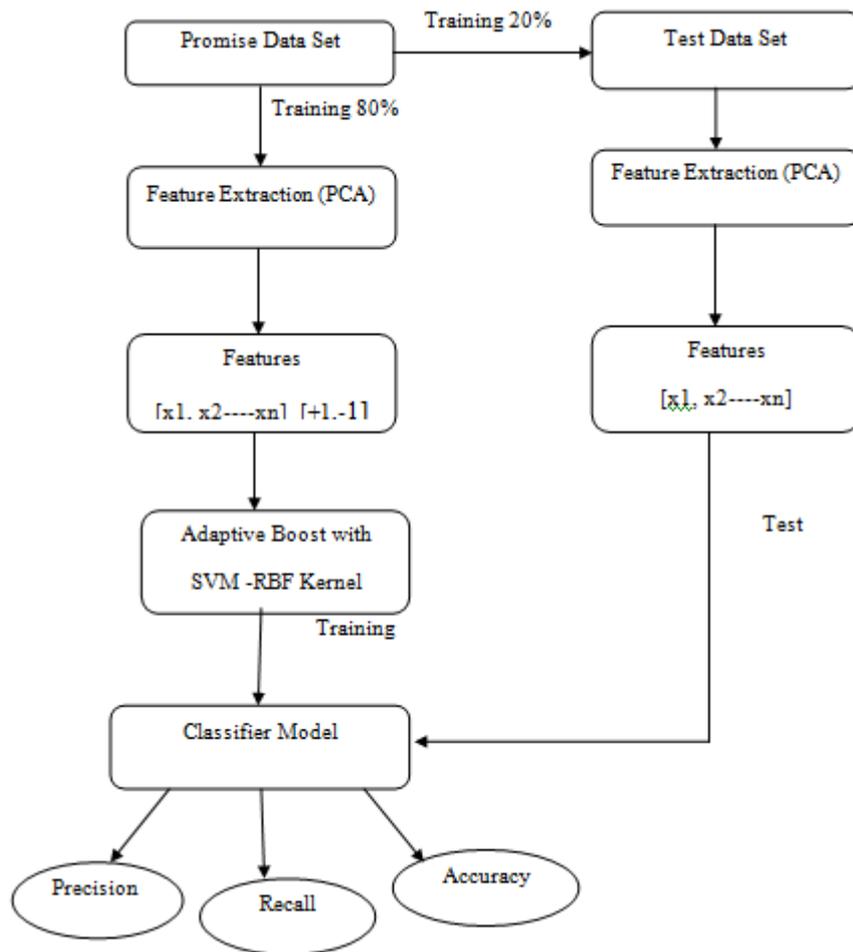
Step 1 : Take the promise data set with 21 different features like cyclomatic complexity, design complexity, effort, time estimator, line count etc for defect prediction in software module.

Step 2 : Implement feature extraction on promise data set by using Principle Extraction Analysis (PCA). Feature Extraction is used to merge the data set. In feature extraction merging process is based on eigen values, having high eigen value means contain more information.

Step 3 : Take the different features $x_1, x_2, x_3, \dots, x_n$ and find out the status that whether they are default or not default [+1, -1]. If the value is +1 that means its 'default' and if -1 then it is 'not default'.

Step 4 : Implement Hybrid Adaptive Boost with SVM -RBF Kernel for component learning and to remove compaction and boundary error condition.

Step 5 : Apply Classifier model to find out precision, recall and accuracy of the software module.



REFERENCES

- [1] Shepperd, Martin, David Bowes, and Tracy Hall. "Researcher bias: The use of machine learning in software defect prediction." *Software Engineering, IEEE Transactions on* 40.6 (2014): 603-616.
- [2] Gray, David, et al. "The misuse of the nasa metrics data program data sets for automated software defect prediction." *Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference on*. IET, 2011.
- [3] Menzies, Tim, Butcher, A., Marcus, A., Zimmermann, T., & Cok, D. "Local vs. global models for effort estimation and defect prediction." *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*. IEEE Computer Society, 2011
- [4] Song, QinbaoJia, Z., Shepperd, M., Ying, S., & Liu, J. "A general software defect-proneness prediction framework." *Software Engineering, IEEE Transactions on* 37.3 (2011): 356-370.
- [5] Czibula, Gabriela, Zsuzsanna Marian, and Istvan Gergely Czibula. "Software defect prediction using relational association rule mining." *Information Sciences* 264 (2014): 260-278.
- [6] Okutan, Ahmet, and Olcay Taner Yildiz. "Software defect prediction using Bayesian networks." *Empirical Software Engineering* 19.1 (2014): 154-181.
- [7] Li, M., Zhang, H., Wu, R., & Zhou, Z. H. "Sample-based software defect prediction with active and semi-supervised learning." *Automated Software Engineering* 19.2 (2012): 201-230.
- [8] <http://rspublication.com/ijst/ijst%20pdf%20feb%2012/18.pdf>
- [9] Data mining , http://en.wikipedia.org/wiki/Data_mining
- [10] Data mining Process, <http://www.google.co.in/images>.
- [11] <http://www.unc.edu/~xluan/258/datamining.html>
- [12] http://www.tutorialspoint.com/data_mining/dm_dti.htm.