1. Name of Researcher: Tripti Lamba

2. Name of Guide: Dr. Kavita

3. Registration Date: 19-Aug-2015

4. Title (with Subtitle): Software Defect Prediction using Data Mining Techniques

5. Alternative Title (if any): NA

6. Abstract

The success of any software system entirely depends on the accuracy of the results of the system and whether it is without any flaws. Software defect prediction problems have an extremely beneficial research potential. Software defects are the major issue in any software industry. Software defects not only reduce the software quality, increase costing but it also suspends the development schedule. Software bugs lead to inaccurate and discrepant results. As an outcome of this, the software projects run late, are cancelled or become unreliable after deployment. Quality and reliability are the major challenges faced in a secure software development process. There are major software cost overruns when a software product with bugs in its various components is deployed at client’s side. The software warehouse is commonly used as record keeping repository which is mostly required while adding new features or fixing bugs. Many data mining techniques and dataset repository are available to predict the software defects. ‘Bug prediction technique’ is an important part in software engineering area for last one decade. Software bugs which detect at early stage are simple and inexpensive for rectifying the software. Software quality can be enhanced by using the bug prediction techniques and the software bug can be reduced if applied accurately. Dependent and independent variable are considered in Software bug prediction. To prevent defect based on software metrics software prediction model are used. Metrics based classification categorize component as defective and non-defective.
The present study was aimed to attain the following objectives:
The main objective of the research was to select and analyze effective software metrics for bug prediction, to select statistical learning and data mining techniques to be applied on historical software data for prediction of software defect and to address various measures to prevent defects in future software version releases.
The Open Science Promise Repository (available at http://openscience.us/repo/defect/ck/), a publicly available dataset was used for this research. The product metrics that is Chidamber and Kemerer metrics suite have been used and the various software modules were taken to discover the most significant or relevant software metrics. The nine software modules taken from CK & OO metrics (Donor-Marian Jureckzo) which were used for the research are named as Ant, Ivy, Tomcat, Berek, Camel, Lucene, POI, Synapse and Velocity. Six CK metrics were WMC, RFC, CBO, LCOM, NOC, DIT and object oriented metrics were CA, CE, NPM, LCOM3, LOC, DAM, MOA, MFA, CAM, IC, CBM, AMC, MAX_CC and AVG_CC. And dependent variable was bug. Software metrics are a very important component in software development area. The most important challenge of any software developer is that ‘the software should be 100% accurate or with minimal defect’ when it reaches to the end-user. The earlier the defect is detected, the earlier the development cost also gets reduced. This is a fact that the more complex the data or software, there is a more probability of the defect. So it is always desirable to use only the relevant and important data for the software development which will definitely reduce time and cost of the developer. More the number of variables, more complex will be the system and more the defects; so it is always preferable to select the small feature set using feature selection and focus only on the important variable. Therefore, it is always desirable to reduce the variables and should include important variables in a dataset. Another important factor of using Feature Selection technique is that if the number of variables is higher than optimal, then the Machine Learning Algorithm exhibits a decrease in accuracy. Feature selection technique can be classified as wrapper, filter and hybrid. Through a Feature Selection technique, we can reduce the variable and locate the importance of the variable in a dataset. For this research,
wrapper and filter method was applied on software modules. The feature selection techniques Boruta, regsubsets and FSelector Random Forest, Information gain, Linear Correlation and Rank Correlation were employed to discover the vital software metrics in a software system. Each of these Feature Selection techniques has derived the most essential and the least essential software metrics. The comparative analysis of feature selection technique was conducted to achieve the most optimal metrics and it was derived that RFC, LOC and WMC were the most optimal metrics and the least significant metrics were DIT and NOC.

Machine learning can be classified as supervised learning and unsupervised learning. Supervised learning can be categorized as Regression and Classification. For the experiment, six regression methods known as Linear Regression, Random Forest, Decision Tree, Support Vector Machine, Neural Network and Decision Stump of machine learning techniques were taken to find out the most optimal model. The dataset was split as 50% training data set and 50% testing dataset, which validates the values of statistical measures for CK and OO metrics. These have been implemented and simulated in R Studio (R.3.3.1) environment. In order to discover best model, comparative analysis was done by using four performance parameters: Correlation, RSquared, Accuracy and Mean square error on six machine learning models. The six machine learning models used were Linear Regression, Random Forest, Decision Tree, Support Vector Machine, Neural Network and Decision Stump on nine software modules such as Ant, Ivy, Tomcat, Berek, Camel, Lucene, POI, Synapse and Velocity using all the CK_OO software metrics and using only optimal software metrics like WMC, RFC and LOC. It was found that the Support Vector Machine is the best model and its accuracy is 83% and mean square error is 0.7 %. Therefore, it is suggested that while developing softwares, more focus should be on RFC, LOC & WMC metrics. In addition, application of Support Vector Machine (SVM) machine learning model would increase the accuracy and reduce the errors.

Further to this, there are various measures to prevent defect in software release. The earlier the defect is detected the cost involved is reduced and the resources
are fully utilized so it becomes much easier to rectify the defect during the initial stage, Complexity metrics are better predictors of fault potential in comparison to other well-known historical predictors of faults, self-review the code has a major contribution in preventing the software defect, files that have anti-patterns tend to have a higher density of bugs than others as anti-patterns can increase the bugs in the future. Anti-patterns can be removed from systems using refactoring.

7. Keywords (at least five): Software metrics, Feature selection, Boruta, regsubsets FSelector, Random Forest, Linear correlation, Rank Correlation, Information gain, Linear Regression, Random Forest, Neural Network, Support Vector Machine, Decision Tree, Decision Stump.