VARIOUS METHODS FOR SOFTWARE FAULT PREDICTION

Shabnam Jariyal[*]

Balraj Singh*

Abstract

Fault estimation of a software module is the probability that the module contain the error and it is a defect which will cause the further failures in the software so the early detection of these errors will help the experts or developers to concentrate on these errors which will improve the quality of software also. As the demand for producing the software quality increases day-by-day, so early estimation of errors in any module is a great step towards it. It will predict the quality attributes like reliability, maintenance, efforts for testing, wrong syntax, wrong logic, misleading documentation, productivity and testing efforts. As the demand for quality of the software increased so the development of the machine learning methods to explore the data are also increased. Early prediction of the faults will make the researchers or experts to give there best in the faulty areas or they can save there much time which is going to be waste in finding the fault prone area in software. These faulty area or modules in the software can cause the failure in extended version. So in this study errors are estimated by using LCOM, BPA, ROC, PCM , DBSCAN.

Keywords- Neural Network, machine learning, Principle component alaysis, Software Fault, Lack of cohesion metric.

[*] Dept. of CSE, Lovely Professional University, Phagwara, Punjab, India

I. INTRODUCTION

As we know the use of software is increased day by day the error or the faults in the software also growing with it and will form a major problem in the future. The fault in the software will cause the failure after its execution. If one can find the faults in early stage then it would help the software developers and testers to save there time, decrease there work load and automatically the cost for it will also decrease. So it is better to estimate the errors in early stage so that the efficiency of the developer will improved. Most of the errors will be made in the design or documentation part or during the implementation part where we do the coding or write the source code. Second reason for early predition is the developers can pay attention to that part of the implementation which will the major cause for the errors.

The faults and errors in the software can be caused by the reasons which are described in below fishbone diagram:

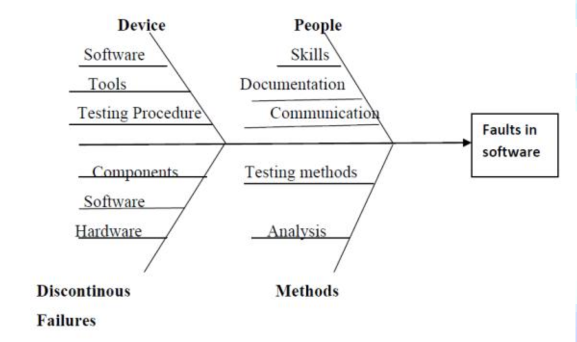

Figure 1.1 Fishbone Diagram of faults in software

II. LITERATURE REVIEW

Yogesh Singh(2010)[1] provide a review different machine learning methods to find out the faults in software. In this six machine learning methods are used to explore the final result. These methods explored to find the effect of static code metrics on software fault prone modules and then compare the result by using ROC analysis(Receiver operating curve). Conclusion shows that the decision tree model is better than other machine learning methods.

K.K.Aggarwal(2007)[2] suggested the effect of design metrics on fault proneness. This explore the relationship between the object-oriented design metrics and the fault proneness of object-oriented system classes. It use data from java applications which contain 136 classes and 26 design metrics. And provide the redundant information. Here the PCM principle component method of analysis is used to find that whether all the metrics are independent or not. Logistic regression analysis is also used to make hypothesis that whether the coupling increase or decrease the fault proneness of a class an after that this will predict which class of java application will be faulty. In this paper the systems under study are medium sized systems written in Java and have a testing record including number of faults found in each class. In this study we first find the interrelationships among selected metrics and then found the individual and combined effect of selected metrics on fault proneness.

Supreet kaur, journal(2012)[3] provide a review of Density-Based spatial clustering of applications with noise (DBSCAN) by which the errors are predicted in java based and C++ language based software. It is a real time assessment technique that classify the system dynamically as error free system. Neural network is used in this and software metric is the input to neural network. This paper shows that using neural network of BP (back propagation) algorithm with the software quality prediction will give the good results.

Saida Benlarbi (1999)[4] suggested the issues in validating object- oriented metrics for the early prediction of risk in software. This paper shows that the empirical validation methods can provide the misleading conclusion about the object oriented metrics. In object oriented metrics we have different classes so it is easy to find the faulty class. This paper tells about the effect of size on different metrics that are used in this paper(WMC, RFC,CBO,LCOM) and by not considering the effect of size. And conclusion will given that the further research will perform by controlling the size.

Lanubile et al. (1995)[5] present the empirical study of modeling techniques to identify the fault in software component at early stage. In this the software complexity measures are used. Techniques used to predict the fault are principal component analysis, logistic regression, discriminant analysis, logical classification model, layered neural networks and holographic networks. This paper describe that no model is there which will able to discriminate between the components which have faults and which will not have faults.

Ruchika Malhotra (2012)[6] provide a review on fault prediction using the statistical and machine learning methods so that the quality of a software can be improved. In this paper one statical and six machine learning methods are applied to predict the model and result is analyzed using AUC (area under curve) obtained from the ROC analysis. It shows that object oriented metrics are useful to predict the fault proneness of classes. Random forest and bagging models give the best result.

III. STUDY OF TECHNIQUES

A. Machine Learning Approach

As to increase the quality of software and the reliability of the software, the early fault prediction technique is the best technique to find the faults. In this the whole data is divided into the different clusters using the clustering techniques like k-means clustering and after that the artificial neural network like RBF(Radial basis function) is applied to the faults predicted [11]. RBF have the better accuracy than the another neural network technique called MLP(multilayer perceptron)

Whereas the study shows that the cost of inspection of MLP is the higher as of the RBF.

A neural network classifer with class label and input metric are used to provide the quality of software by the use of median- adjusting class labels. It include the MLP, three multilayer analysis and a set of metrics with class labels.

An effective preprocessing method to predict the quality of software is called median-adjusting class labels [13]. As in case of object oriented software system the MLP technique is used to predict the faults. This is done on the basis of classification, as the faults are classified , these are again classified according to there type of fault [12]. So from above study the MLP can be used to predict the faulty classes and the RBF is used to categorize these classes based on there type.

When we have a small number of data set and want to maintain the software quality SVM (support vector machine) model is used. It is a technique developed for the data set classification. As we have several techniques to predict the faults, the SVM is the most robust in nature, with good efficiency and gives the better performance [14].

We have number of techniques included in statistical models and machine learning models but no such models give the best accurate results for data sets.

The object oriented class metrics are used in the BPN (back propagation neural network) and PNN (probalistic neural network) for the fault prediction in software [15]. In this they collected the data set of the scholastic institution and then compare the results of above two neural network technique with the statistical methods using the five quality parameters [15]. After the study they show that the PNN is much robust and better technique than the BPN.

In order to minimize the cost and to improve the efficiency of the software, they use the SVM (Support vector machine) and ANN(Artificial neural network) on the data set obtained from

NASA data repository [16]. We have so many software metrics like McCabe and Halstead which use the SVM and ANNs to classify the faulty modules.

B. Statistics Based Technique

To perceive faulty components in software during the early stage of the software development and to make a error free model using the object oriented classes statistic based techniques are used in wide range. The design quality of the prediction model is low and the accuracy of these are high. These model collect the data from the commercial java application and will identify the faults for the inconvenience in future release of software application.

In order to compare the accuracy of the fault prediction models as available before the system is implemented and after its implementation, the design metrics [17] and code metrics [18] are used. Design metrics are only available before the coding will start whereas the code metrics is available after the development of system. Models are developed using the linear regression and data is taken from the telecommunication system which is developed by the Ericsson. There study shows that the prediction made after the development of system is 34 % more accurate than before the development of the system. The unevenness of the metrics obtainable after and before the implementation is 58% and 43% [19] and when these metrics are not used the performance of the system will remain the same.

C. Statistical and Machine Learning Techniques

In order to predict the faults in an open source software system, the research team use the statistical techniques and machine learning techniques [20]. According to it the performance and precision of LOC (lines of code) and LCOM (lack of cohesion on methods) is of good quality. They compare the statistical methods with machine learning methods to find the impact on static code metrics. They use the ROC (Receiver Operating characteristics) curve to determine the performance by calculating the AUC(area under curve) from the ROC [1].

D. Expert Estimation Technique

The expert estimation are limited for the large systems. Statistical models discussed above are cheap to build and they perform well for small and large scale systems also. Statistical methods can be applied without the experts also as discussed in [15].

E. K-Nearest Neighbor Technique

There are so many sampling techniques which are used conventionally but these were use the class attributes not the non class attributes [22] and this is the reason that they use J48 for NASA datasets. The dataset is divided into the three main groups:

- Training set
- Nice neighbor test sets
- Nasty neighbor test sets

The result shows that the nasty test is 20% accurate and the nice test is 94% accurate.

The KNN (k-nearest neighbor) [23] method is used for clustering and classification. The classification techniques based on KNN are as follows:

- DB-KNN (The Density Based KNN Classifier)
- V-KNN (The Variable K nearest Neighbor Classifier)
- W-KNN (The Weighted KNN Classifier)
- CB-KNN (The Class Based KNN Classifier)
- D-KNN (The Discernibility KNN Classifier)

The above methods are discussed [23], in this they concluded that DB-KNN method is slower than KNN because of its structural density but on the other hand the performance is good and also a reliable classifier. V-KNN and W-KNN classifier has good performance and fast than KNN. CB-KNN is the more accurate than all other.

The KNN is divided into two categories : structure less and structure based . these categories are used for the different purposes, the structure less is used to overcome the memory limitations and the structure based is used to reduce the complexity.

IV. CONCLUSION

The outcome of the above study is that one can effectively identify the defects to improve the quality of software. and can achieve the high software reliability. From the above study it is concluded that the machine learning methods are better to estimate the faults then the statistical methods.

V. PROPOSED WORK

A GUI must be created which will calculate the parameters which are responsible for creating the faults in software like complexity, short changed quality assurance, lack of user support, uncontrolled employee problems and many more. User can select from these parameters which he want to consider while calculating the error or fault in application software. And after that k-means clustering is applied to data set obtained from selected parameters. So by this we can classify the fault prone area. One can work on more parameters also.

VI. References

[1] Yogesh singh and Arvinder Kaur (2010) “ Prediction of Fault-Prone Software Modules using Statistical and Machine Learning Methods.” , 2010 international journal of computer applications (0975-8887) volume 1- No.22.

[2] K.K.Aggarwal, Ruchika Malhotra (2007) “Investigating effect of design Metrics on fault proneness in object-oriented System “ journal of Object technology, JOT, 2007 Vol. 6, N0.10.

[3] Supreet Kaur and Dinesh Kumar (2012) “ Software Fault prediction in object oriented software systems using density based clustering approach”. , international journal of research in engineering and technology (IJRET) Vol. 1 No.2 , ISSN: 2277-4378.

[4] Saida Benlarbi (1999) “Issues in validating object-oriented metrics for early risk prediction”, FastAbstract ISSRE

[5] Fenton, N.E. (1999) “ A critique of software defect prediction models” , IEEE 1999, centre for software reliability, London.

[6] Ruchika malhotra and Ankita jain (2012) “Fault prediction using statistical and machine learning methods for improving software quality” , journal of information processing system. Vol.8, No2.

[7] Yan Ma Guo, L. (2006), “A Statistical Framework for the Prediction of Fault-Proneness”, West Virginia University, Morgantown.

[8] Lanubile (1995) “Comparing models for identifying fault-prone software components”.

[9] Catal, Cagatay,"Software fault prediction: A literature review and current trends", Expert Systems with Applications 2010.

[10] Parvinder S.Sandhu, Jagdeep Singh, Vikas Gupta(2010) “A K-Means Based Clustering approach for finding faulty Modules in Open source Software system”, world academy of science, Engineering and technology 48 2010.

[11] Mahaweerawat, A, Sophasathit, P, Lursinsap, C., "Software fault prediction using fuzzy clustering and radial basis function network" International conference on intelligent technology , Vietnam.

[12] Mahaweerawat, A, Sophatsathit, P, Lursinsap, C, Musilek,P "Fault prediction in object-oriented Software using neural network techniques" Advanced Virtual and Intelligent Computing Center, Chulalongkorn University,Thailand,, 2004.

[13] Pizzi Summer "Software quality prediction using median-adjusted class labels" 2002 International Joint Conference on IEEE, 2002.

[14] Xing, Lyu,"A novel method for early Software quality prediction based on support vector machine In Software Reliability Engineering, 2005. 16th IEEE International Symposium on IEEE.

[15] Kanmani,V.Rhymend Uthariaraj, " Object-oriented Software fault prediction using NN" Information and software technology 2007.

[16] Gondra Iker ,"Applying machine learning to Software fault proneness Prediction" Journal of System and Software 2008.

[17] El Emam, Khaled, Walcelio Melo, "The prediction of faulty classes using object-oriented design metrics", Journal of systems and Software, No. 1, 2001.

[18] Zhao, Ming, C. Wohlin,"A comparison between Software design and code metrics for the prediction of software fault content", Information and Software Technology 40, No. 14, 1998.

[19] Tomaszewski, P., Lundberg, "The accuracy of early fault prediction in modified code", In Proceedings of the Fifth Conference on Software Engineering Research and Practice in Sweden (SERPS).

[20] Gyimothy, Tibor,"Empirical validation of object-oriented metrics on open source Software for fault prediction", Software Engineering, IEEE Transactions on 31, No. 10, 2005.

[21] Boetticher, Gary D.,"Nearest neighbor sampling for better defect prediction", In ACM SIGSOFT Software Engineering Notes, Vol. 30, No. 4, 2005.

[22] Voulgaris, Z., Magoulas, G. D.,"Extensions of the k nearest neighbour methods for classification problems.International Conference on Artificial Intelligence and Applications, Austria, February .