A survey – Quality based Object Oriented Software Fault Prediction

R.Sathyaraj[#], S.Prabu[*]
School of Computing Science and Engineering
VIT University
Vellore-14
[#]sathyaraj@vit.ac.in, [*]sprabu@vit.ac.in

Abstract:

Software fault prediction is the most efficient methodology to improve the quality of the products. To improve the quality it is essential to find the error or fault as quick as possible. To determine and advance the development of product there are different prediction approaches are available like correction cost prediction, test effort prediction, software fault prediction (SFP), security prediction and so on. This paper survey the variety of methods and metrics used to improve the quality of the object oriented software. We compared different prediction models and proposed the methodology with the datasets they used for easy understanding. Software fault prediction using different techniques to improve the quality and the error free software delivery. This paper gives overview about the prediction models for software in design and implementation phase and also it discuss trends which are currently used software fault prediction.

Index Terms: **Software quality, SFP- Software fault prediction, fault proneness, quality metrics**

I. INTRODUCTION

Software engineering is the profession to analysis, design, development and maintenance of software. Traditional software development methods focus on software correctness, including performance issues in the development process. The success of the software depends on the quality, which measures how good the software is designed and also the requirement design meet the final output. Also it is related with cost effective, mean time failures and security predictions.

Errors, faults, failures and defects are connected terms. An error is a human mistake, it causes wrong output. Error makes the software failure which means software not meets its specified requirements. This failure causes defect.

Software quality is the main aspect to predict the fault in object oriented software systems. It is achieved by different types of models proposed in software verification and validation.

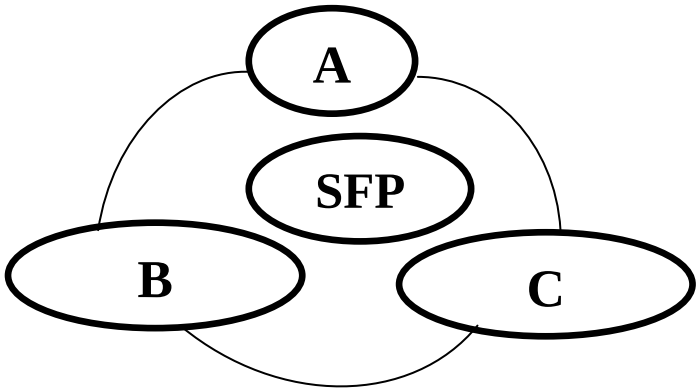

Figure1: Route map of software fault prediction

A – Design Phase
B – Implementation Phase
C – Testing Phase

Naïve Bayes is based on the Bayes' theorem with naïve postulation. It measured probability of the product, regardless of the features may present or absent. This algorithm calculates the number of cases of one property and second property occurs together then divides with number of cases where first property can occur alone.

Literature Survey:

Cagatay Catal (2011) gives the detailed literature review and current trends in software fault prediction. This paper discussed about metrics, methods, datasets and results of the previous studies regarding software fault prediction. The problem occurs to predict the software fault when the previous software fault data was missing and if it may be the new project. To overcome this problem X-means clustering-based method used, also fuzzy clustering and k-means clustering-based methods used in the experiments. But the system effecting cost of metrics collection tools are expensive, so need influential model to predict fault with limit fault data. In this paper, showed the Naive Bayes algorithm is best for semi-supervised fault prediction models with limited datasets [1].

Taghi m.Khoshgoftaar and Naeem Seliya (2003) compared six fault prediction techniques they are, CART-LS (Classification and regression trees tools – least square), CART-LAD (Classification and regression trees tools – Least absolute deviation), S-PLUS (S language –based statistical modeling tool), Multiple linear regression, Artificial neural networks, case based reasoning. In this paper they collected software metrics from four releases of very large telecommunications systems. To find the accuracy of different prediction models they used metrics like performance, average absolute value and average relative errors. Model created using RAW original software metrics and their principle components (PCA).

Prediction models are treated as treated as factor. Blocking variables formed by system release and model type i.e. RAW or PCA. Here they used parallel comparison for average absolute error and average relative error and tabularize the performance of the model. Finally CART –LAD model performs the better way when compared to all other models. S-Plus model was ranked as sixth position. Comparison results shows the order of the prediction models are, (from first to sixth position)

CBR < MLR < ANN < CART-LS < S-PLUS [2].

Taghi m.Khoshgoftaar, Kehan Gao and Robert M.Szabo (2001) used Zero-inflated poisson regression model and Poisson regression to predict fault in the system and make the system to reliable. This paper investigates the full-scale industries software systems. Here this Poisson regression model used to undervalue the zeros of dataset variable values. The file metrics includes the length and number of source codes, comments length in source code. Conclusion of comparing the two models prove that the zero-inflated Poisson regression provide better performance [3].

Supreet Kaur, and Dinesh Kumar (2012) describes the java based object oriented software system fault prediction performance is based on density based spatial clustering of application with noise (DBSCAN). So DBSCAN is evaluated java based object oriented systems with accurate value. In this paper, for prediction they used metric based approach. Also the performance of fault proneness of classes is evaluated for C++ language based components which is based on NASA metrics data program data repository. Here discussed about metrics based approach is good in the reduced set of attributes, it is achieved from the results clearly. This describes that with reduced set of attributes the accuracy of the prediction is increased from 85.3712% to 90.6114%. DBSCAN result showed the accuracy of prediction of 58.63% for the reduced set attributes. In both, density based clustering provides probability detection for non-faulty model equal to 1 and false alarm probability for faulty model is equal to 0. So its shows DBSCAN are acceptable for prediction based on fault proneness [4].

Andrian Marcus and Denys Poshyvanyk proposed a measure for object oriented software systems by increasing the cohesion of classes and less coupling between methods. Unstructured information which is embedded in the source code is used for this process. Comments and identifiers are the unstructured information. The measure C3 (conceptual cohesion of classes) is used for fault prediction in object oriented software systems. To determine the cohesion of the class different ways were there like structural metrics, semantic metrics and so on. Here to extract comments and identifiers in object oriented systems Information retrieval approach was used with the help of latent semantic indexing. This paper describes the fault prediction with case study work. In the first case study discussed principle component analysis of the metric data. Next case study shows the fault prediction in classes. Finally the results shown that conceptual cohesion of classes C3 is a precious complement in a number of combination with other structural cohesion metrics based on the regression analyses results. It promotes the prediction of software system [5].

Saida et al (1999) investigated that as early as possible the predictors find the fault in the classes otherwise the maintenance cost will be increased. This is the principle stay back in the development of object oriented metrics [6].

Denaro, Lavazza, and Pezzè (2003) proposed fault prediction in an industrial telecommunication system. They used logistic regression in telecommunication system via class level metrics. In this learning by using automated tool can't calculate the CBO and LCOM metrics, so the metrics anticipate coupling between classes and in methods there is a lack of cohesion. Compare to lines of code metrics, multivariate models not offer any advantages and also they describes for fault proneness none of the metrics is connected. This study is

implemented first in C and involved with an object oriented systems. This evolution and bad OO design causes correlated problem. Combination of a header and cpp file is a module in this [7].

CONCLUSION:

This paper describes the way to improve the quality of software with the methodologies and concepts used to imagine the software fault prediction. This paper also discussed the restriction to focus during fault prediction. From this survey, we can understand the importance of fault prediction and helps to develop a new or combination of these any techniques to produce a better software fault prediction.

REFERENCES:

[1] Cagatay Catal., "Software fault prediction: A literature review and current trends", Expert systems with applications Volume 38, Issue 4, April 2011, Pages 4626-4636 Elsevier.

[2] Taghi m.Khoshgoftaar and Naeem Seliya (2003), "Fault Prediction Modeling for Software Quality Estimation: Comparing Commonly Used Techniques", in Empirical Software Engineering, 8, 255–283, 2003.

[3] Khoshgoftaar. T, Gao.K and Szabo.R.M, "An application of zero-inflated poisson regression for software fault prediction", in 12th international symposium on software reliability engineering, Washington, DC: IEEE Computer Society, PP 66–73, 2001.

[4] Supreet Kaur, and Dinesh Kumar, "Software Fault Prediction in Object Oriented Software Systems Using Density Based Clustering Approach", in International Journal of Research in Engineering and Technology (IJRET) Vol. 1 No. 2 March, 2012 ISSN: 2277-4378.

[5] Marcus.A, Poshyvanyk.D, and Ferenc.R., "Using the conceptual cohesion of classes for fault prediction in object-oriented systems", IEEE Transactions on software engineering, vol. 34, no. 2, March/April 2008.

[6] Saida Benlarbi,Khaled El Emam, Nishith Geol (1999), "Issues in Validating Object-Oriented Metrics for Early Risk Prediction", by Cistel Technology 210 Colonnade Road Suite 204 Nepean, Ontario Canada K2E7L5.

[7] Denaro, G., Lavazza, L., & Pezzè, M (2003), "An empirical evaluation of object oriented metrics in industrial setting", in the 5th CaberNet plenary workshop, Porto Santo, Madeira Archipelago, Portugal.