EasyChair Preprint

№ 6246

Software Fault Prediction Using Classification Algorithm

Pranita Ingale and Nilesh Alone

August 6, 2021

Software Fault Prediction Using Classification Algorithm.

Pranita S.Ingale
Department Of Computer Engineering
G.E.S. R.H. Sapat College of Engineering, Nashik
pranitaingale96@gmail.com

Prof. Nilesh V.Alone
Department Of Computer Engineering
G.E.S.R.H.Sapat Collage Of Engineering, Nashik
nilesh.alone@gmail.com

ABSTRACT

Software fault prediction is a valuable exercise in software quality assurance to best allocate the limited testing resources.Software testing is a crucial activity during software development and fault prediction models assist practitioners herein by providing an upfront identification of faulty software code by drawing upon the machine learning literature. Previous research on software metrics has shown strong relationships between software metrics and faults in object-oriented systems using a binary variable. Practically, it would be helpful if developers could identify the most error-prone modules early so that they can optimize testing-resource allocation and increase fault detection effectiveness accordingly. The findings can provide an effective foundation for managing the necessary activities of software development and testing.

***Keywords*—Software fault prediction, Bayesian networks, classification, Fault distribution, open source software, Pareto principle, software quality, software reliability, software testing.**

I) INTRODUCTION

One of the most important aspects of software development and project management is how to make predictions and assessments of quality and reliability for developed products.Software fault prediction plays the important role in designing the different software fault prediction models for fault prediction. Software fault prediction models have been broadly classified into many models such as Kernel Regression Models, Linear Regression Models, Zero-inflated Poisson Models. Each software fault prediction models have their own application with different behaviour to perform their task. Mostly used software fault prediction models are: Linear Regression Models(LR), Zero-inflated Negative Binomial models (ZINB), Zero-inflated prediction model in software data, hybrid model reconstruction for cross-project defect prediction approach (HYDRA) , empirical studies of a two-stage data pre-processing approach, learning-to-rank approach to software defect prediction and fuzzy rule-based approach for software fault prediction. This paper, discusses five different software fault prediction approach such asPerformance tuning for automotive software fault prediction [1], An empirical analysis on effective fault prediction model developed using Ensemble methods [2],Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Software [3],Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers [4],Empirical study of fault prediction for open-source systems using the Chidambaram and Kemmerer metrics [5].An Empirical Study On Clustering Approach Combining Fault Prediction[6],Software Defect Prediction based on Geometric Mean for Subspace Learning [7],Software Fault Prediction Based On One-Class SVM [8],An Investigation of Essential Topics on Software Fault-Proneness Prediction classification [9],Artificial neural network-based metric selection for software fault-prone prediction model [10].
These software fault prediction approaches provide the better effectiveness, improved performance, better predictability, accuracy and quality. But these methods also have some problem so to overcome such problems improve version of software fault prediction models that is “Design model using classification algorithm For Software fault prediction” software fault prediction model is proposed here that depend upon the decision tree classification algorithm.

II) BACKGROUND

As per the studies on software fault prediction many models and approaches have been develop for the software fault predictions in recent past years. Such approaches are:

Performance tuning for automotive software fault prediction which proposes to apply a wide range of under and oversampling parameters to determine the achievable performance and influences upon seven classification algorithms[1].An Empirical analysis on effective fault prediction model developed using Ensemble methods is developed for predicting fault proneness and to validate the source code metrics and select the right set of metrics with objective to improve the performance of fault prediction model [2].Bounded Generalized Pareto Analysis which proposed to fix the issue, and can provide a more accurate prediction and analysis of open-source software (OSS) fault distributions[3].Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers approach is presented that Augmented Naive Bays classifiers can yield similar or better performance than the commonly used Naive Bays classifier[4].Fault prediction for open-source systems using the Chidambaram and Kemmerer metricsApproach has proposed to use the past incidents of faults to rate classes into several categories: low, medium and high. These ratings are tested to build several fault-prediction models using many data mining tools[5].

This paper introduces Ten software fault prediction models i.e. Performance tuning for automotive software fault prediction, An Empirical analysis on effective fault prediction model developed using Ensemble methods, Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Software, Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers, Empirical study of fault prediction for open-source systems using the Chidambaram and Kemmerer metrics.These are organizes as follows. **Section I** Introduction. **Section II** discusses Background. **Section III** discusses previous work. **Section IV** discusses existing methodologies. **Section V**discusses attributes and parameters and how these are affected on software fault prediction models. **Section VI** proposed method and outcome result possible. Finally **section VII** Conclude this review paper.

III) PREVIOUS WORK DONE

In research literature, many software fault prediction models have been studied to provide various fault prediction schemes and improve the performance in terms of fault prediction, effectiveness, software reliability, accuracy, software testing.

Harald Altingeret al. [1] have worked on Performance tuning for automotive software fault prediction that gives the higher performance values but the various classifiers are influenced in different ways.This proposed model uses historic data from the same project for training and prediction

Lov Kumar et al.[2] has proposed An Empirical analysis on effective fault prediction model developed using Ensemble methods for predicting fault proneness and that validates the source code metrics and select the right set of metrics with objective to improve the performance of fault prediction model

Chin-Yu Huanget al.[3]has worked on Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Software, approach for Software Fault Prediction that outlines plans to further investigate whether or not the fault distributions of OSS can still be accurately described and analyzed by the PD model.

Karel Dejaeger et al.[4] has presented Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers approach to construct simpler networks with fewer nodes and arcs remain unexplored for software fault prediction.

Raed Shatnawiet al. [5] have proposed Empirical study of fault prediction for open-source systems using the Chidambaram and Kemmerer metrics model for software fault prediction. In this, a dependent variable is proposed that uses fault history to rate classes into four categories (none, low risk, medium risk and high risk) and to improve the predictive capability of fault models.

Leo Xiai al. [6] has work on adopting clustering algorithm which have the best cluster number to test case prioritization whether could improve the efficiency of regression testing.

Yan Gao al. [7] has work on Software Defect Prediction based on Geometric Mean for Subspace Learning model, GMCRF (Geometric Mean Conditional Random Field) method based on dimensionality reduction technique and CRF Model is introduced to improve software defect prediction in imbalance distribution by choosing the best combination of features for data set and incorporating complex features without any process in training stage.

Lin Chen al. [8] has presented Software Fault Prediction Based On One-Class SVM, aims at solving the problem that the faulty samples are too rare to train a classifier, an one-class SFP model is proposed by using only non-faulty samples based on one-class SVM.

Shou-Yu Lee al. [9] has work on An Investigation of Essential Topics on Software Fault-Proneness Prediction that represents an investigation of essential topics in this area, including techniques for evaluating the effectiveness of fault-proneness prediction models, issues of concerns when building the prediction models, as well as findings shared by the academic community.

C. Jin al. [10] have proposed Artificial neural network-based metric selection for software fault-prone prediction

model, authors proposed a reduction dimensionality phase, which can be generally implemented in any software fault-prone prediction model. In this, the authors present applications of artificial neural network (ANN) and support vector machine in software fault-prone prediction using metrics.

IV) EXISTING METHODOLOGIES

Many software fault prediction approach have been implemented over the last several decades. There are different methodologies that are implemented for different software fault prediction models i.e. Performance tuning for automotive software fault prediction, An empirical analysis on effective fault prediction model developed using Ensemble methods, Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Software, Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers, Empirical study of fault prediction for open-source systems using the Chidambaram and Kemmerer metrics.

1] Performance tuning for automotive software fault prediction: Performance tuning for automotive software fault prediction has presented a new model for software fault prediction. This proposed model developed uses historic data from the same project for training and prediction, defined as 'within-project' SFP. To overcome this author applied a wide range of under and oversampling parameters to determine the achievable performance and influences upon seven classification algorithms. The original dataset is split in test and training data prior any modification, using the train data on the sampling approach and the test data to evaluate the performance[1].

2] An empirical analysis on effective fault prediction model developed using Ensemble methods: An empirical analysis on effective fault prediction model developed using Ensemble methodswhich addresses three different ensemble methods to develop a model for predicting fault proneness. The A framework is proposed to validate the source code metrics and select the right set of metrics with the objective to improve the performance of the fault prediction model. The fault prediction models are then validated using a cost evaluation framework. Ensemble method learning algorithm outperforms individual classifiers [2].

3] Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Software: Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Softwareapproach for Software Fault Prediction that is outline plans to further investigate whether or not the fault distributions of OSS can still be accurately described and analyzed by the PD model. In this paper, the potential issues of using the classical PD model for the prediction of software fault distributions was first investigated. The proposed BGPD model was shown to eliminate the issues which occurred in the classical PD model, and to exhibit great performance on modeling the distribution of software faults. Parameters of all models were estimated by using three different methods: LSE, MLE, and MOM. Some experiments were performed and discussed in detail based on real OSS failure data. Numerical results showed that the proposed BGPD model is applicable to accurately describe the distribution of software faults in a complex system. Finally, early fault distribution analysis is also studied, and experimental results provide useful information that can be instrumental to various management decisions for the project manager and development team [3].

4] Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiersapproach: Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers approach tries to answer the issues by comparing 15 Bayesian network learners both in terms of the Area Under the ROC Curve and the recently introduced H-measure. The results of the experiments show that Augmented Naive Bays classifiers can yield similar or better performance than the commonly used Naive Bays classifier. This additional performance, however, comes at the expense of more complex models. Considering comprehensible models only, Augmented Naive Bays classifiers using the Local Leave-One-out Cross Validation quality measure are to be recommended. The Naive Bays classifier, which can be turned into a linear model, is also a valid alternative, despite its simple network structure. General Bayesian Networks were found to be either outperformed by other Bayesian learners or to result in overly complex network structures. It can be argued that networks which focus on a smaller set of highly predictive features provide practitioners with the means to gain insights more easily into the drivers of software faults and, to further capitalize hereon, the use of MB feature selection was also tested. The outcome indicates that NB is able to reduce the number of variables while not negatively impacting performance. However, other feature selection approaches are possibly able to select an even smaller set of highly predictive features[4].

5] Empirical study of fault prediction for open-source systems using the Chidambaram and Kemmerer metricsApproach: Empirical study of fault prediction for open-source systems using the Chidambaram and Kemmerer metricsApproach for Software Fault Prediction, the proposed work uses Chidambaram and Kemmerer metrics to quantify the risks of having faults in future releases to focus limited resource son significant problems that face the project. In this approach, the

history of faults is considered to improve the reliability of fault predictions by rating faulty classes into three ratings (low, medium and high) and by reducing the imbalance in fault distributions. This rating results in a new dependent variable that is compared with the usually considered classification of classes into faulty and not faulty. The objective of this empirical study is to analyze the use of fault history in building fault-prediction models in the open-source field. The proposed model aim to use class history to rate classes into s several areas. To validate the research objectives, author use the evolution of many open-source systems. The quality of these systems is measured using a set of metrics. This method uses seven machine learning classification techniques that were widely used to build fault-prediction models. Finally, author statistically tests the significance of the difference between the proposed models (multi-category classification) and traditional models (binary classification)[5].

6] An Empirical Study On Clustering Approach Combining Fault Prediction For Test Case Prioritization: Clustering Approach Combining Fault Prediction for Test Case Prioritization implement a new prioritization technique incorporating clustering algorithm and code fault prediction. Author investigate adopting clustering algorithm which have the best cluster number to test case prioritization whether could improve the efficiency of regression testing. In this both test cases prioritization in each cluster and cluster prioritization considered, the cluster prioritization depend on the result code fault prediction based SVM algorithm, and test cases in each cluster prioritization depend on the distance between test case and centre point [6].

7] Software Defect Prediction based on Geometric Mean for Subspace Learning: In this Software Defect Prediction based on Geometric Mean for Subspace Learning model, GMCRF (Geometric Mean Conditional Random Field) method based on dimensionality reduction technique and CRF Model is introduced to improve software defect prediction in imbalance distribution by choosing the best combination of features for data set and incorporating complex features without any process in training stage. GMCRF can easily be trained using the simple direct optimization technique of stochastic gradient descent. This GMCRF method achieves much better final results than the other approach [7].

8] Software Fault Prediction Based On One-Class SVM: Software Fault Prediction Based On One-Class SVM, aims at solving the problem that the faulty samples are too rare to train a classifier, an one-class SFP model is proposed by using only non-faulty samples based on one-class SVM. The empirical validation is conducted on 6 extremely imbalanced datasets collected from real-world software containing only small amounts of faulty instances. The test results suggest that the proposed model can achieve a reasonable fault prediction performance when using only a small proportion of training sample and performs much better than conventional and class imbalanced learning based SFP models in terms of G-mean measure [8].

9] An Investigation of Essential Topics on Software Fault-Proneness Prediction: An Investigation of Essential Topics on Software Fault-Proneness Prediction that represents an investigation of essential topics in this area, including techniques for evaluating the effectiveness of fault-proneness prediction models, issues of concerns when building the prediction models, as well as findings shared by the academic community. In order to build a fault-proneness prediction model, a multitude of factors need to be considered during model construction. In order to reduce the risk of software faults manifesting during operation, techniques are needed to identify code which has the potential to cause problems early on so that more effort can be spent on testing to prevent such problems from occurring. This is the reason that stimulates the proposal of various fault-proneness prediction models. Author presented an objective to offer a fast and convenient index not only for professionals to efficiently identify their issues of concern in software fault-proneness prediction but also for researchers [9].

10] Artificial neural network-based metric selection for software fault-prone prediction model: Artificial neural network-based metric selection for software fault-prone prediction model, authors proposed a reduction dimensionality phase, which can be generally implemented in any software fault-prone prediction model. In this, the authors present applications of artificial neural network (ANN) and support vector machine in software fault-prone prediction using metrics. A new evaluation function for computing the contribution of each metric is also proposed in order to adapt to the characteristics of software data. The vital characteristic of this approach is the automatic determination of ANN architecture during metrics selection. It is also very simple because its implementation requires neither extra cost nor expert's knowledge. The proposed model has good performance, and can provide software project managers with trustworthy indicators of fault prone components [10].

V) RESULT AND DISCUSSION

Performance tuning for automotive software fault predictionuses wides range of under and oversampling parameters to determine the achievable performance and influences upon seven classification algorithms[1]. An Empirical analysis on effective fault prediction model developed using Ensemble methods approach shows how to validate the source

code metrics and select the right set of metrics with objective to improve the performance of fault prediction model[2]. Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Software model shows that how to eliminate the issues which occurred in the classical PD model and to exhibit great performance on modeling the distribution of software faults [3]. Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiersshows the Augmented Naive Bays classifiers can yield similar or better performance than the commonly used Naive Bays classifier[4]. Empirical study of fault prediction for open-source systems using the Chidambaram and Kemmerer metricsuses seven machine learning classification techniques that were widely used to build fault-prediction models. In this work, a dependent variable is proposed that uses fault history to rate classes into four categories (none, low risk, medium risk and high risk) and to improve the predictive capability of fault models[5].

Prediction models and approach	Advantages	Disadvantages
I. Performance tuning for automotive software fault prediction	This proposed method provide testing effort is very high and restrictive development processes ensure high quality bug data information.	In this proposed method When using under- and oversampling one needs to carefully choose the classifier and decide whether to favors precision Or recall.
II. An Empirical analysis on effective fault prediction model developed using Ensemble methods	This method improve the performance of the fault prediction model. The fault prediction models are then validated using a cost evaluation framework.	The drawback of this method is cross validation cannot be used in practice.
III. Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Software	This proposed method provide better performance and predictability than other models during the early stage (40% and below), middle stage (50% and 60%), and late stage (80% and above) of testing.	The drawback of this method need to improve the checklists, hold more training seminars, and use more powerful and useful computer-aided software engineering (CASE) tools
IV. Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers	This method provide better performance than the commonly used Naive Bays classifier.	This method have more cost.
V. Empirical study of fault prediction for open-source systems using the Chidambaram and Kemmerer metrics	In this approach, new dependent variable is proposed to improve the quality of the data sets. The results of classifiers have improved for the later releases as the software matures.	This method need to tackle common problems that affect the effectiveness of software prediction models such as data imbalance and noise reduction.

TABLE 1: Advantages And Disadvantages Of SFP Models.

VI) PROPOSED METHODOLOGY

Software fault prediction approach is important and difficult task to analyse and discuss about various methods based on different parameters i.e. accuracy, quality, cost, time, flexibility, effectiveness, etc for different software fault prediction models. There are still problems which trouble in this field. New software fault prediction method called " Analysis of various software fault prediction models and design model using classification algorithm for Software fault prediction" for more effective and more accurate fault prediction model is propose here to overcome the problems of previous models..The proposed method uses decision tree classification algorithm that is easier as compare to other methods.

A. Classification Analysis :

Data Classification is a form of analysis which builds a model that describes important class variables. For example, a model built to categorize bank loan applications as safe or risky. Classification methods are used in machine learning, and pattern recognition.

Decision Tree is used to build classification and regression models. It is used to create data models that will predict class labels or values for the decision-making process. The models are built from the training dataset fed to the system (supervised learning).

Using a decision tree, we can visualize the decisions that make it easy to understand and thus it is a popular data mining technique.

B. Basic concept of Classification Algorithm:

Data Mining: Data mining in general terms means mining or digging deep into data which is in different forms to gain patterns, and to gain knowledge on that pattern. In the process of data mining, large data sets are first sorted, then patterns are identified and relationships are established to perform data analysis and solve problems.

Classification: It is a Data analysis task, i.e. the process of finding a model that describes and distinguishes data classes and concepts. Classification is the problem of identifying to which of a set of categories (subpopulations), a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.

Decision Tree algorithm:

1. Place the best attribute of the dataset at the root of the tree. Internal nodes of the decision nodes represent a test of an attribute of the dataset leaf node or terminal node which represents the classification or decision label.
2. Split the training set into subsets. Subset should be made in such a way that each subset contains data with the same value for an attribute.
3. Repeat step 1 and 2 on each subset until you find leaf nodes in all the branches of the tree.

Diagrammatic representation of proposed method is shown as follows:

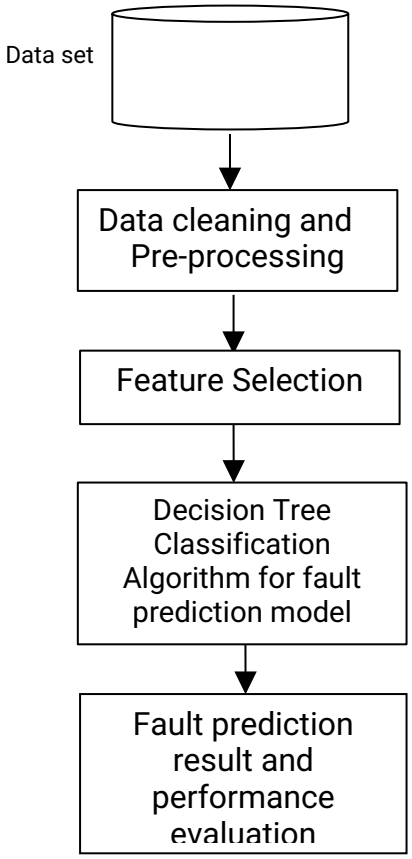

Fig 1: Block diagram of proposed SFP model using classification algorithm

The working of proposed method is as shown as above figure 1 which shows the step by step process.

VII) CONCLUSION

This paper focused on the study of various mobility scheme i.e. Performance tuning for automotive software fault prediction, An Empirical analysis on effective fault prediction model developed using Ensemble methods, Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Software, Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers, Empirical study of fault prediction for open-source systems using the Chidambaram and Kemmerer metrics.But there are some problems in accuracy, performance and quality of

software data so to improve this software fault prediction model using classification algorithm is proposed here.

ACKNOWLEDGMENT

I hereby take this opportunity to thank G.E.S. R.H. Sapat College of Engineering, Nashik for providing the opportunity to showcase my capabilities and skills. I would like to thank my guide, Prof. N.V. Alone, for guidance, support and valuable inputs. Also I would take this opportunity to express my heartfelt gratitude towards the people who helped me in presenting the paper directly or indirectly

REFERENCES

[1] Harald Altinger, Steffen Herboldy, Friederike Schneemann, Jens Grabowskiy, and Franz Wotawaz Gaimersheim,"Performance tuning for automotive software fault prediction", *SANER,* 2017.

[2] Lov Kumar, Santanu Rath, Ashish Sureka"An Empirical analysis on effective fault prediction model developed using Ensemble methods*",IEEE 41ST ANNUAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE*, 2017.

[3] Chin-Yu Huang, Chin-Song Kuo, and Shao-Pu Luan, "Evaluation and Application of Bounded Generalized Pareto Analysis to Fault Distributions in Open Source Software",*IEEE TRANSACTIONS ON RELIABILITY*, VOL. 63, NO. 1, MARCH 2014.

[4] Raed Shatnawi, "Empirical study of fault prediction for open-source systems using the Chidamber and Kemerer metrics", IET Software, Vol. 8, Iss. 3, 2014.

[5] Karel Dejaeger, Thomas Verbraken, and Bart Baesens,"Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers", *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, Vol. 39, NO. 2, FEBRUARY 2013.

[6] Lei Xiao, Huaikou Miao, Weiwei Zhuang, Shaojun Chen, "An Empirical Study On Clustering Approach Combining Fault Prediction For Test Case Prioritization", IEEE Computer Society, 5090-5507, *Pg No.* 815 - 820, *MAY 2017.*

[7] Yan Gao, Chunhui Yang, Lixin Liang, "Software Defect Prediction based on Geometric Mean for Subspace Learning", *IEEE,* 4673-8979, *Pg No.* 225-229, AUGUST 2017.

[8] Lin Chen, Bin Fang, Zhaowei Shang, "Software Fault Prediction Based On One-Class SVM", International Conference on Machine Learning and Cybernetics, 5090-0390, *Pg No.*1003-1008, *JULY, 2016.*

[9] Shou-Yu Lee, Dong Li, Yihao Li, "An Investigation of Essential Topics on Software Fault-Proneness Prediction", International Symposium on System and Software Reliability (ISSSR), 5090-5563, *Pg No. 113-119, AUGUST, 2016.*

[10] C. Jin, S.-W. Jin, J.-M. Ye, "Artificial neural network-based metric selection for software fault-prone prediction model", IET Software, Vol. 6, Iss. 6, *Pg No.* 479–487, *MAY 2012.*