Journal of Engineering Sciences

Vol 13, Issue 03, MARCH /2022

ISSN NO:0377-9254

Software Fault Prediction and Analysis using Machine Learning: Engineering College Prospective

Prof. Jyotsnarani Tripathy-Faculty at Gandhi Institute For Technology, CSE Dept. (Affiliated to Biju Patnaik University of Technology)
Dharmendra Chik Baraik – Student at Gandhi Institute For Technology, CSE Dept. (Affiliated to Biju Patnaik University of Technology)
Atanu Ghosh – Student at Gandhi Institute For Technology, CSE Dept. (Affiliated to Biju Patnaik University of Technology)
Jyotiranjan Mallik – Student at Gandhi Institute For Technology, CSE Dept. (Affiliated to Biju Patnaik University of Technology)

ABSTRACT

In the software engineering community, software defect prediction and analysis is a critical issue. Software fault prediction can directly affect the quality and has achieved significant popularity in the last few years. This software prediction analysis helps in delivering the best development and makes the maintenance of software more reliable. This is because predicting the software faults in the earlier phase improves the software quality, efficiency, reliability and the overall cost in SDLC. Developing and improving the software defect prediction model is a challenging task and many techniques are introducing for better performance. The comparison is made with other machine learning algorithms to finds the algorithms which gives more accuracy. And the results show that machine learning algorithms gives the best performance. The existence of software defects affects dramatically on software reliability, quality, and maintenance cost. Achieving reliable software also is hard work, even the software applied carefully because most time there is hidden errors. In addition, developing a software defect prediction model which could predict the faulty modules in the early phase is a real challenge in software engineering. This is because anticipating issues prior to programmed launch increases user satisfaction while also improving overall software performance. Furthermore, early detection of software flaws promotes software adaption to various settings and maximizes resource consumption.

KEYWORDS: software defect prediction; software fault prediction; mobile application; review; systematic literature review; deep learning; machine learning.

I.INTRODUCTION

Software fault prediction is one of the important aspects to be considered while developing a software. It helps to make the system more reliable. Amongst the other software predictions such as cost prediction, security prediction and others, fault prediction is the most important and the reason it has been researched the maximum through all these years. Having a fault prediction model helps in cost efficiency and more importantly time efficiency and also adds to the quality assurance of the software. The measures to tackle the faulty modules can be prepared beforehand in the scenario of any problem. It is very helpful in the development of a larger software where the probability and the frequency of faults can be more. Fault prediction helps in the maintainability of the software. Software fault tolerance on the other hand, comes into play after the fault has occurred. It ensures the continuous

working of the software after the fault , adding to the reliability and increasing the dependability of the system. It adds to the ability of working properly even when some of the internal component goes to a failure state. Fault can happen due to any design issue, functionality error or code error. The tolerance of the errors happening at the runtime is more than the compile time errors. Fault tolerant system are getting more importance today as they ensures the no-halt service mechanism. If the faults are not dealt with for long, then the consequences can be major and may disturb the actual outcome of the software. Various techniques have been proposed to tackle Software Bug Prediction (SBP) problem. The most known techniques are Machine Learning (ML) techniques. The ML techniques are used extensively in SBP to predict the buggy modules based on historical fault data, essential metrics and different software computing techniques.

II.SCOPE OF WORK

Software fault localization and maintainability are defined as a software system or modules can be adapted to correct faults, improve performance or software and system testing, software development techniques or modify to a changed platform. A software defect predictive model enables organizations to help to reduce the maintenance effort, time and cost overall on a software project. To ensure the quality of good software must be reliable, and it can occur a smaller number of failures during the software run time . Hence, classification of defects on software modules has a large impact during software development process. But the real scenario would become hard, because when developer changes his program inside an application and it is related to other modules including failed to updated version of this application. Therefore, it is very possible case for the software become faulty and not stable . The number of literatures in software fault proneness is increasing day by day for demand for automated services.

III. LITERATURE SURVEY

This paper states that there are only a few software modules that causes faults in the software system. That is the reason that only a few are with faulty labels and others with non-faulty labels, such datasets are termed as “imbalanced” and there are various methods through which we can evaluate their performance. The methods that are widely used are evaluation metrics; different researchers have used different evaluation parameter. Hence, it is very difficult to compare and contrast the work that is done in present to that which is done in the past. The researchers have used various evaluation parameters until now, and to select a common parameter for evaluation is critical in important in software engineering today.

In this paper, four supervised ML learning classifiers are used to evaluate the ML capabilities in SBP. The study discussed Naïve Bayes (NB) classifier, Decision Tree (DT) classifier, KNN classifier and Support Vector Machine (SVM) classifier.

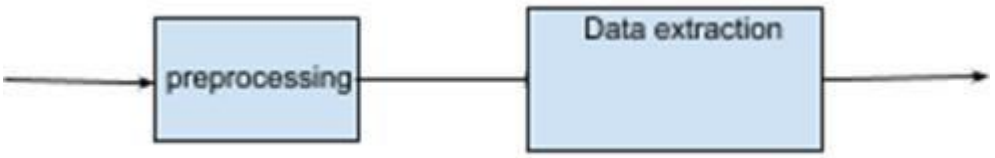
This research paper discusses about the fault tolerance happening the application layer. This type of fault tolerance means to detect and prevent the software faults in the application layer only because such faults are not handled by the operating system or the hardware layer of the system. We take only the flaws which cause the application procedure to hang or crash; they incorporate application software faults and the errors which go undetected in the other layers of hardware and operating system.

IV.OBJECTIVE

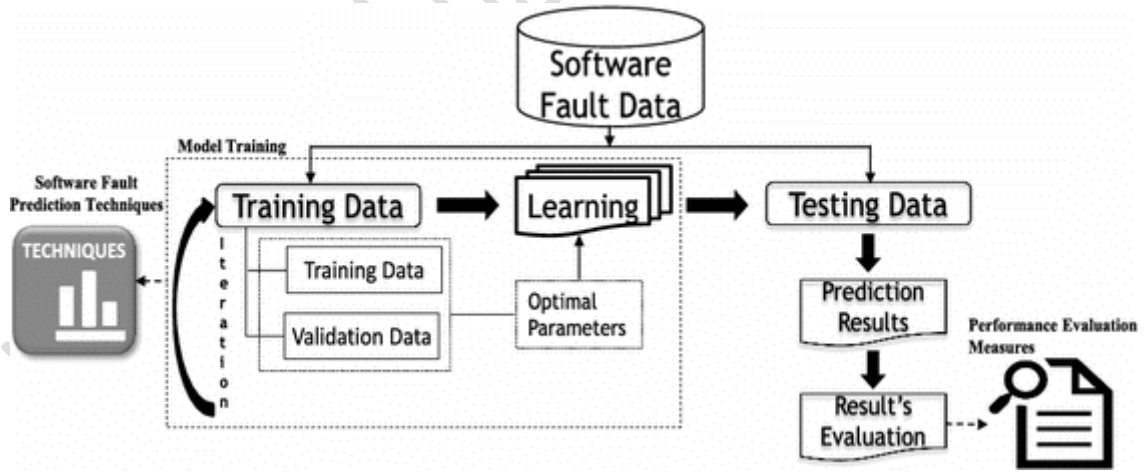
Software defect prediction models are commonly used to detect faulty software modules based on software metrics collected during the software development process. Objective: Data mining techniques and machine learning studies in the fault prediction software context are mapped and characterized. We investigated the metrics and techniques and their performance according to performance metrics studied. An analysis and synthesis of these studies is conducted. Method: A systematic mapping study has been conducted for identifying and aggregating evidence about software fault prediction.

V. DATA PREPROCESSING

It is not always the case that we come across clean and prepared data wheworking on a machine learning project. It is also necessary to clean and format data before doing any action with it. As a result, we employ the data preparation task. We need to do the pre processing of the data so that the features of the data can be extracted. After preprocessing load the data into the next step.



VI. ARCHITECTURE DIAGRAM



VII. USED MACHINE LEARNING ALGORITHMS

The study aims to analyze and assess four supervised Machine Learning algorithms, which are Naïve Bayes (NB), K-Nearest Neighbour (KNN), Support Vector Machine (SVM) and Decision Tree

(DT). The study shows the performance accuracy and capability of the ML algorithms in software bug prediction and provides a comparative analysis of the selected ML algorithms.

The supervised machine learning algorithms try to develop an inferring function by concluding relationships and dependencies between the known inputs and outputs of the labeled training data, such that we can predict the output values for new input data based on the derived inferring function. Following are summarized description of the selected supervised ML algorithms:

- Naïve Bayes: NB is an efficient and simple probabilistic classifier based on Bayes theorem with independence assumption between the features. NB is not single algorithms, but a family of algorithms based on common principle, which assumes that the presence or absence of a particular feature of the class is not related to the presence and absence of any other features.
- Decision Tree: DT is a common learning method used in data mining. DT refers to a hierarchal and predictive model which uses the item's observation as branches to reach the item's target value in the leaf. DT is a tree with decision nodes, which have more than one branch and leaf nodes, which represent the decision.
- Support Vector Machine(SVM): SVM is a directed learning strategy. SVM gains from preparing the dataset and it is utilized for grouping. On the off chance that we think about a lot of preparing models, each case having a place with one of two classes, an SVM calculation assembles a model that aides in anticipating whether the model falls into one class or the other one.
- K-Nearest Neighbour(KNN): K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

  K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm

VIII. CONCLUSION

Software Defect Prediction can directly affect the quality and has achieved significant popularity in the last few years. This software prediction analysis helps in delivering the best quality product without any defects. Therefore this helps in deploying the products that are error free. Here we performed this using machine learning algorithms Naive Bayes, Support vector machine, Decision Tree and KNN. When we observed the accuracies obtained between these algorithms Decision Tree is more accurate than Naive Bayes, KNN and SVM.

IX. FUTURE WORK

Future scope in software fault prediction became one of the noteworthy research topics since 1990, and the number of research papers is almost doubled until year 2009. Many different techniques were used for software fault prediction such as genetic programming, decision trees neural network, Naïve Bayes, case-based reasoning, fuzzy logic and the artificial immune recognition system algorithms in. Menzies et al. have conducted an experiment based on public NASA datasets using several data mining algorithms and evaluated the results using probability of detection, probability of false alarm and balance parameter. They used log-transformation with Info-Gain filters before applying the algorithms and they claimed that fault prediction using Naïve Bayes performed better than the J48 algorithm. They

also argued that since some models with low precision performed well, using it as a reliable parameter for performance evaluation is not recommended.

REFERENCES

[1]. Malhotra, R. An empirical framework for defect prediction using machine learning techniques with Android software. Appl. Soft Comput. 2016, 49, 1034–1050.
[2]. Misirli, A.T.; Bener, A.B. A mapping study on Bayesian networks for software quality prediction. In Proceedings of the 3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, Hyderabad, India, 3 June 2014; pp. 7–11.
[3]. Kaur, A. A Systematic Literature Review on Empirical Analysis of the Relationship between Code Smells and Software Quality Attributes. Arch. Compute. Methods Eng. 2019, 27, 61267–61296.
[4]. Alsolai, H.; Roper, M. A systematic literature review of machine learning techniques for software maintainability prediction. Inf. Softw. Technol. 2020, 119, 106214.

[5]. I. C. Society, "IEEE 729-1983 - IEEE Standard Glossary of Software Engineering Terminology," 1982.

[6]. "Naive Bayes," Wikipedia.

[7]. "Support Vector Machine," Wikipedia.

[8]. "Decision Tree," Wikipedia.

[9]. "KNN," Wikipedia.

[10]. "NASA Promise Dataset Repository".

Vol 13, Issue 03, MARCH /2022

ISSN NO:0377-9254