IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-7, Issue-2, Feb 2019 E-ISSN: 2347-2693

Application of KNN Classification Technique in Detection of Software Fault

Ritika[1*], Er. Saurabh Sharma[2]

[1,2]Dept. of Computer Science Engineering, Sri Sai University Palampur, HP, India

[*]*Corresponding Author: rc60447@gmail.com*

DOI: https://doi.org/10.26438/ijcse/v7i2.389393 | Available online at: www.ijcseonline.org

Accepted: 16/Feb/2019, Published: 28/Feb/2019

Abstract--- The software engineering is the technology which is used to analyze software behavior SDP includes software metrics, their attributes like line of code etc. The main goal of software defects prediction model includes ordering new software modules based on their defect-proneness and classifying them whether it is new software or not. The main purpose of SDP for the ranking is to predict which modules have the most defects to define software quality enhancement. The goal of SDP for the ranking task is to predict the relative defect number, although estimating the precise number of defects of the modules is better than estimating the ranks of modules, because the precise number of defects can give more information than the ranks. The software defect prediction technique is applied in the previous work based on the technique of ANN. In this research work the technique of KNN is applied for the software defect prediction. It is analyzed that proposed technique has high accuracy and less execution time as compared to existing ANN technique.

Keywords---Fault Prediction, KNN, Software Defect Prediction, NFR ,ANN

I. INTRODUCTION

A collection of programs, procedures, data and documentation is known as software. By keeping in consideration the hardware and operating system which is also known as a platform, the software is designed for a particular organization. A systematic method through which the software is developed is known as engineering. The designing of software is highly complex which thus needs various guidelines. Defect can be simply defined as a variance between expected and actual [1]. Defect is a slip-up found once the appliance goes into production. It commonly refers to several troubles with the software products, with its external behavior or with its internal features. In other words Defect is the difference between expected and actual result in the context of testing. In different words defect is that the distinction between expected and actual end in the context of testing. It is the deviation of the customer requirement. Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime [2].It varies from organization to organization and additionally from project to project because it is ruled by the computer code testing method and additionally depends upon the tools used. There's no one right thanks to rely on package quality –its sophisticated area.It is helpful, however, to cluster its varied parts into three broad aspect .The three aspects of package quality area unit practical quality, structural quality, and method quality. SDP includes software metrics, their attributes like line of code etc [3]. The main goal of software defects prediction model includes ordering new software modules based on their defect-proneness and classifying them whether it is new software or not. First of all, data is obtained from old software modules with known defect numbers and including values of all software modules with known defect numbers. It included software values of the software metrics and their attributes. Furthermore, model based approach like machine learning to construct model and obtained data from them. The proposed model predicts the number of defects of new modules. An order has been received according to the prediction model and allocates resources according to the testing order. The main purpose of SDP for the ranking is to predict which modules have the most defects to define software quality enhancement [4]. The goal of SDP for the ranking task is to predict the relative defect number, although estimating the precise number of defects of the modules is better than estimating the ranks of modules, because the precise number of defects can give more information than the ranks. To construct a prediction model, we have a tendency to should have defect and activity information collected from actual software package development efforts to use because the learning set. There exist compromise between however well a model fits to its learning set and its prediction performance on extra information sets. Therefore, we should evaluate a model's performance by comparing the predicted defectiveness of the modules in a test set against their actual

defectiveness. Several defect prediction techniques have been proposed by researchers. The artificial neural network relies on the human biological system in its design and style. It processes information in a way similar to the human brain using an intricate system of interconnected neurons to solve highly complex problems [5]. Genetic Algorithms is an approach to machine learning that behaves equally to the human factor and also the Darwinian theory of activity. Genetic Algorithms square measured enforced starting with a personal population that's typically drawn within the sorts of trees. The Fuzzy Logic model relies on the idea or reasoning and works on a worth that's approximate in nature. It is a intensify from mathematical Boolean Logic wherever there will solely be True or False. The problem of classification has been wide studied within the information, data processing, and knowledge retrieval communities. The training information is employed so as to construct a classification model, that relates the options with in the underlying record to one of the category labels. For a given instance of test that the category is unknown, the training model is employed to predict a category label for this instance. In the onerous version of the classification problem, a selected label is expressly alloted to the instance, whereas within the soft version of the classification issue, a chance worth is appointed to the checking instance. Other alterations of the classification task permits ranking of various class selections for a testing instance, or enable the designation of multiple labels to a testing instance [6]. The application of Support vector machine (SVM) method is to Text Classification. The SVM need each positive and negative training set uncommon for different classification ways. A neural network classifier could be a network of units, wherever the input units typically represent terms, the output unit(s) represents the class. For classifying a document under test, its term weights are allotted to the input units; the turning on of those units is propagating forward through the network, and also the price that the output unit(s) takes up as a consequence determines the categorization call. When decision tree is employed for text classification it consist tree internal node area unit labeled by term, branches outward from them that labeled by test on the weight, and leaf node are represent corresponding class labels. Naïve bias method is kind of module classifier under known priori probability and class conditional probability .it is basic idea is to calculate the probability that document D is belongs to class C.

II. LITERATURE REVIEW

In the software development projects a major key challenge role is played by the requirement analysis. In the software projects, the involvement of the specific customer requirements and management has various impacts. Hence, it is required to improve this area more in terms of both academic and industrial fields. They presented the model of CMMI in order to expose the development and management requirement. It also specifies the various different goals and practical platform for them as well [7]. They listed in this paper the requirement of the management and key challenges and issues faced by it with the help of the model CMMI and its regular activity. The technical documentation plays an essential role in determining the success and failure of any software for which a Software requirement specification document has been utilized. To resolve this author in this paper proposed an effective approach [8]. In this method, four processes were involved. On the basis of the received concern from stakeholders, RMM was generated to minimize ambiguities and incorrectness. The requirements of the client and SRS have been checked by the inspection of the third party. A detailed report to team of requirement engineers was submitted by the third party after performing detailed analysis using inspection models and assigning total quality score. Development of the high quality software is the process based on the requirements of the software in which each step is related to SR. This technique requires human effort so that created features can be analyzed easily [9]. They proposed the use of deep learning in order to classify the requirements of the software without utilizing the feature engineering. On the basis of the Convolutional Neural Network this proposed method also stated as art in other languages. They utilize the PROMISE corpus for the evaluation of the proposed method in which a set of labeled requirements in functional and different categories of nonfunctional requirements was utilized. On the basis of the performed experiments, it is concluded that proposed method shows effective results for the SRC using CNN. An essential role is played by the safety system in the automotive sector. These safety systems are ABS brake systems, stability control system, and air bag systems. Among these safeties system one is oriented towards collisions which are activated at different interval of time with objectives. For instance, in order to prevent the impact of the collision, these systems are activated [10]. The objective of minimizing the physical harm to occupants these systems is activated during collision. The time between the collision and arrival of outside assistance for the vehicle's occupant's minimization is the main objective after the collision. Some of the recommendations of the requirements are satisfied by the presented SRS in the automotive standard ISO 26262 and to the requirements of the standard ISO/IEC/IEEE 15288 they utilized the processes in this paper. In the management of the software project an essential role is played by the estimation of the software effort in which requirements is its major driver. In order to identified the which non-functional requirements are used they performed the literature review so that their effects and which method is used can be identified easily [11]. Obtained result, demonstrated that non-functional requirements has been utilized by the 33% of 39 algorithmic methods. Hence, these published results were not definite when the correlation with estimation accuracy was investigated. They also conducted the quasi-experiment on

publicly available datasets to address the issue. Therefore, experimental results show that there is reduction in the 30% in the estimation error with the use of nonfunctional requirements. In this paper that within the electronic systems, the reverse engineering is performed for various reasons. In this paper, the imaging parameters of a successful tomography are also presented here along with the combination of advanced 3-D image processing. The associated time and cost of these systems is reduced by providing an analysis to automate RE. The two PCBs which are a four-layered custom design board and the complex commercial board are presented in this paper. On the basis of these experiments, the performances of proposed algorithms are analyzed and it is seen that there is an improvement in the proposed methods in comparison to the traditional approaches [12].

SDP models play a very crucial role in the identification and detection of faults in different modules of software. Some of these models are used for quality improvement[13].In this paper they proposed that defects in software may result in degradation of the software quality which will be cause of software failure .It is very important to make efforts for minimizing defects in software. The software defect prediction model construction can benefit from directly optimizing the model performance measure for ranking task have been discussed in paper [14].For predicting defects in software several classification technique have used like SVM, Decisions trees, Random forest, Naïve bayes, ANN. SVM classification technique used for the defect prediction in [15][16],Decision tree algorithm is employed for the same effort in[17][18][19] and Naïve Bayes [20] [21][22].ANN Algorithm has used for the problem in [23][24].

III. RESEARCH METHODOLOGY

The software fault prediction is the technique which can predict the faults. The pre-training ANN technique is applied for the software prediction. In this research work, KNN classification technique will be replaced with the ANN technique for the software prediction and then for detection. The KNN is the classification technique which is applied to classify similar and dissimilar data into more than one class. K-Nearest neighbor classifiers depend on learning by analogy. The training samples are depicted by n dimensional numeric attributes. Every sample represents a point in an n-dimensional space. Along these lines, the greater part of the training samples is stored in an n-dimensional pattern space. At the point when given an unknown sample, a k-nearest neighbor classifier looks the pattern space for the k training samples that are closest to the unknown sample. "Closeness" is defined in terms of Euclidean distance. Not at all like decision have tree induction and back propagation, nearest neighbor classifiers assigned break even with weight to every attribute. This may bring about confusion when there are numerous irrelevant attributes in the data. Nearest neighbor classifiers can likewise be utilized for prediction, that is, to give back a genuine valued prediction for a given unknown sample. For this situation, the classifier gives back the average value of the genuine valued associated with the k nearest neighbors of the unknown sample. The k-nearest neighbors' algorithm is among the simplest of all machine learning algorithms.

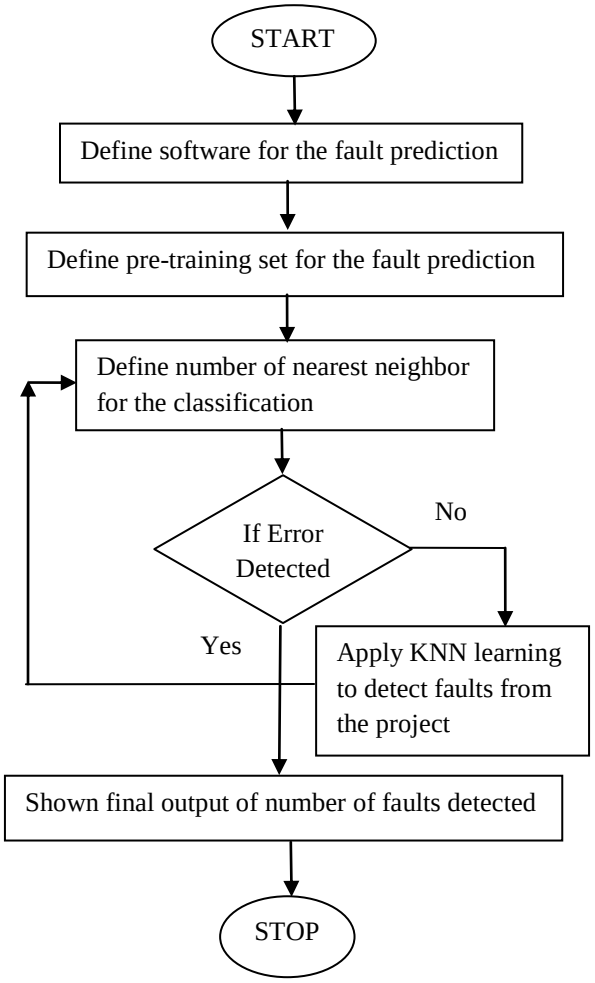

Fig 1: Flowchart of Software fault Detection Process

IV. EXPERIMENTAL RESULTS

The proposed research has been implemented in Python and the results are evaluated by making comparisons against proposed and existing research in terms of certain performance parameters. The datasets are given in .csv format with comma as the delimiter and point as decimal mark. The datasets were cleared of files that contained ".example" or ".tests" in their path. A complete description of features inside the datasets is given in Table of Metrics.

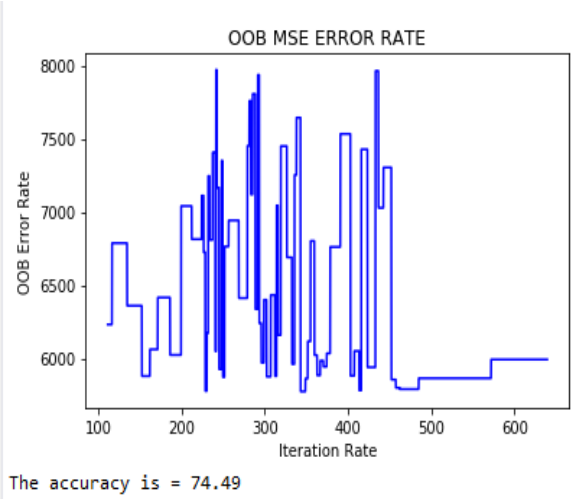

Fig 1: Error Rate of ANN technique

As shown in figure 1, the ANN technique is executed for the fault prediction in the software. The accuracy of fault prediction 74.5 percent

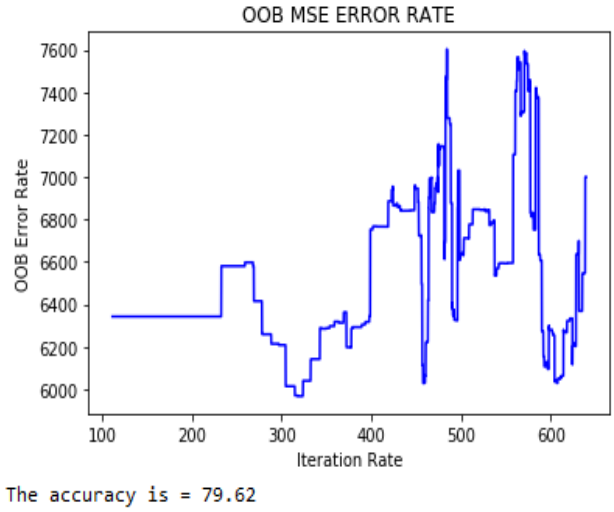

Fig 2: Error Rate of KNN technique

As shown in figure 2, the KNN technique is executed for the fault prediction in the software. The accuracy of fault prediction 79.62 percent.

$$\text{Accuracy} = \frac{\text{correctly classified defects}}{\text{all data}} * 100$$

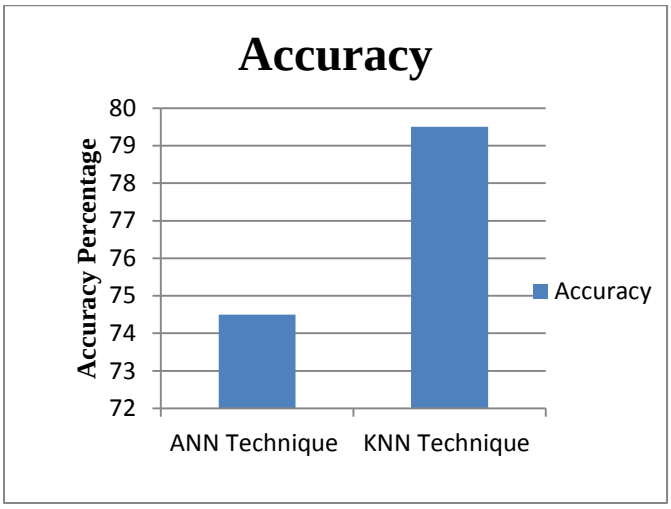

Fig 3: Accuracy Comparison

As shown in figure 3, the accuracy of proposed method is compared with the existing method. It is analyzed that accuracy of proposed method is high as compared to existing due to use of KNN with NFR matrix

Table 1: Accuracy Comparisons

TECHNIQUE	ACCURACY
ANN	74
KNN	79.6

Table1. shows the accuracy comparision of previous ANN technique with the proposed KNN technique table shows the KNN technique is having higher accuracy i.e. 79.6 as compared to ANN technique i.e.74.

V. CONCLUSION & FUTURE SCOPE

In this research work, the KNN classification technique has been used for the detection of faults in software. For detection of bugs in software testing procedure includes comparison of the expected outcome with the actual outcome which helps in detecting the errors which may occur during the development of the project. In the previous work ANN technique was used for fault prediction.In this research work, the ANN will be replaced with the KNN classification technique for the software defect prediction with more accuracy. The performance of KNN technique is analyzed in terms of accuracy and execution time and compared with the previous ANN technique. It is analyzed that KNN technique having more accuracy than ANN technique, it performs well in terms of all defined parameters i.e.. For continuation of this work ,we will suggest to use other deep learning models for improving the accuracy and performance.

REFERENCES

[1] Dang Viet Dezung, Atsushi Ohnishi Improvement of Quality of Software Requirements with Requirements Ontology", International Conference on Quality Software, Jeju, pp 284- 289, 2009

[2] F. Shull, et al., How Perspective-Based Reading Can Improve Requirements Inspections, IEEE Computer, Vol. 33, No. 7, pp. 73-79, Jul.2000.

[3] C. Denger and T. Olsson, Quality Assurance in Requirements Engineering, in Engineering and Managing Software Requirements, A. Aurum and C. Wohlin, Eds. Springer, pp. 163-185, 2005

[4] J. G. Hall, M. Jackson, R. C. Laney, B. Nuseibeh, and L. Rapanotti, "Relating software requirements and architectures using problem frames," in Proc.of the IEEE Joint International Conference on Requirements Engineering, 2002, pp. 137–144.

[5] P. Grnbacher, A. Egyed, and N. Medvidovic, "Reconciling software requirements and architectures: The cbsp approach," in Fifth IEEE International Symposium on Requirements Engineering, 2001, pp. 202–211.

[6] M. Brandozzi and D. Perry, "Transforming goal oriented requirement specifications into architectural prescriptions," in Proceedings of First International Workshop From Software Requirements to Architectures (STRAW01), 2001, pp. 54–61.

[7] Senay Tuna Demirel, Resul Das, "Software Requirement Analysis: Research Challenges and Technical Approaches", IEEE, 2018

[8] Syed Waqas Ali, Qazi Arbab Ahmed, Imran Shafi, "Process to Enhance the Quality of Software Requirement Specification Document", IEEE, 2018

[9] Ra´ul Navarro-Almanza, Reyes Ju´arez-Ram´ırez, Guillermo Licea, "Towards supporting Software Engineering using Deep Learning: A case of Software Requirements Classification", 2017 5th International Conference in Software Engineering Research and Innovation

[10] Jorge Rafael Aguilar Cisneros, Genaro de la Rosa Garcia, Carlos Alberto Fernández-y-Fernández, "Software Requirement Specification for the Automotive Sector: The case of a Post-Collision Event Control System", 2017 5th International Conference in Software Engineering Research and Innovation

[11] Stallin E. F. Silva, Mario L. Cˆortes, "Use of non-functional requirements in software effort estimation: systematic review and experimental results", 2017 5th International Conference in Software Engineering Research and Innovation

[12] Navid Asadizanjani, Mark Tehranipoor, and Domenic Forte, "PCB Reverse Engineering Using Nondestructive X-ray Tomography and Advanced Image Processing", 2017, IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY

[13] Mrinal Singh Rawat, Sanjay Kumar Dubey , "Software Defect Prediction Models for Quality Improvement: A Literature Study", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, September 2012 ISSN (Online): 1694-0814

[14] Xiaoxing Yang, Ke Tang, Senior Member, IEEE, and Xin Yao, "A Learning-to-Rank Approach to Software Defect Prediction", IEEE TRANSACTIONS ON RELIABILITY, VOL. 64, NO. 1, MARCH 2015

[15] B. Twala, "Software faults prediction using multiple classifiers," in: 2011 3rd Int. Conf. Comput. Res. Dev., 4, 2011.

[16] L. Yu, "An evolutionary programming based asymmetric weighted least squares support vector machine ensemble learning methodology for software repository mining," Inf. Sci. 191, 2012.

[17] L. Guo, Y. Ma, B. Cukic, H.S.H. Singh, "Robust prediction of faultproneness by random forests," in: 15th Int. Symp. Softw. Reliab. Eng., 2004.

[18] A. Kaur, R. Malhotra, "Application of random forest for predicting fault prone classes," in: International Conference on Advanced Computer Theory and Engineering, Thailand, December 20–22, 2008.

[19] T. Khoshgoftaar, E. Allen, "Model software quality with classification trees," Recent Adv. Reliab. Qual. Eng., 2001.

[20] A. Bener, B. Turhan, "Analysis of Naive Bayes' assumptions on software fault data: an empirical study," Data Knowl. Eng. 68, 2009.

[21] K. Dejaeger, T. Verbraken, B. Baesens, "Prediction Models Using Bayesian Network Classifiers," 39, 2013.

[22] A. Okutan, O.T. Yıldız, "Software defect prediction using Bayesian networks," Empirical Softw. Eng., 2012.

[23] T. Khoshgaftaar, E.D. Allen, J.P. Hudepohl, S.J. Aud, "Application of neural networks to software quality modeling of a very large telecommunications system," IEEE Trans. Neural Netw. 8 (4), 1997.

[24] A.T. Mısırlı, A.B. Bener, B. Turhan, "An industrial case study of classifier ensembles for locating software defects," Softw. Qual. J. 19, 2011.

