# Computation Intelligence Techniques for Software Defect Prediction: A Review

## Vijay Kumar Boudh[1], Prabhat kumar Bharti[2], Deena Nath[3]

[1]M. Tech Student, Department of Computer Science & Engineering, Goel Institute of Technology & Management, Lucknow

[2]Assistant Professor, Department of Computer Science & Engineering, Goel Institute of Technology & Management, Lucknow

[3]Assistant Professor, Department of Computer Science & Engineering, Goel Institute of Technology & Management, Lucknow

---***---

**ABSTRACT:-** *The software defects at the early stage of development, may lead to degradation of the quality which might be the underlying cause of failure of that software. High quality of software is safeguarded by Software reliability and Software quality assurance. Both these concepts are drawn in throughout the software development and maintenance process. The activities like the performance analysis, functional tests, quantifying time and budget along with measurement of metrics are used to ensure quality. A software bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from behaving as intended (e.g., producing an incorrect result). Further, Software fault prediction facilitates to software engineers to pay attention to development activities on defect less code which enhance the software quality and minimize the cost and time to develop software system in today's era. There are many prediction models which are used to filter the software defects. This paper surveys literature review of articles for the past many years in order to explore how various prediction methodologies have been developed during this period in order to take care of the issues related to software defect.*

**Keywords:** *Software defect, Fuzzy Logic, ANSIF*.

## 1. INTRODUCTION

A software fault is a defect that causes software failure in an executable product. In software program engineering, the non- conformance of software program to its necessities is typically known as a malicious program. Most bugs arise from mistakes and errors made by using individuals in either a program's layout or its data source code, a few are because of compilers generating incorrect code. Knowing the causes of feasible defects as well as figuring out trendy software program manner regions that may need interest from the initialization of a venture may want to save money, time and work. The possibility of early estimating the capacity faultiness of software program could assist on planning, controlling and executing software program development sports.

Developing a disorder free software device may be very difficult and maximum of the time there are some unknown insects or unexpected deficiencies even in

software tasks in which the principles of the software improvement methodologies had been carried out cautiously. Due to a few faulty software modules, the renovation phase of software projects could come to be without a doubt painful for the users and luxurious for the businesses. That is why, predicting the defective modules or documents in a software machine previous to undertaking deployment is a totally vital activity, because it leads to a lower inside the overall price of the mission and a growth in general mission fulfilment price. Defect prediction will give one extra chance to the improvement team to retest the modules or documents for which the defectiveness probability is high.

To gain software excellence all through improvement outstanding significance needed to apply to the following 3 sports specially: Defect prevention; Defect detection; Defect correction. Thus, so that it will address the above noted problems. An strive has been made inside the present work for software program errors prediction.

## 2. LATEST RESEARCH TRENDS

Since past many years' organizations have been seeking answer to the question that how to forecast the quality issue of their softwares before its final utilization. In order to tackle this issue many journals have given details about the metrics and statistical techniques.

**Xiaoxing Yang, et al.** used the rank overall performance optimization approach for software forecasting version improvement. For this rank to mastering technique become used [16]. The version was developed on preceding paintings and changed into later studied for enhancing the overall performance of the model. The work includes two components: one is a unique application of the learning-to-rank method to real-world data sets for software disorder prediction, and the opposite is a comprehensive assessment and comparison of the learning-to- rank approach towards different algorithms that have been used for predicting the order of software modules according to the expected quantity of defects. This have a look at suggests that the effect of optimization of the model overall performance using rank to gaining knowledge of technique clearly improve the prediction accuracy.

**Mrs. Agasta Adline, Ramachandran. M** predicting the fault-proneness of application modules whilst the fault labels for modules are unavailable is a tough venture frequently raised in the software industry [15]. They tried to expect the fault–proneness of an application modules when fault labels for modules aren't present. Supervised strategies like Genetic set of rules-based software fault prediction method for type has been proposed.

**Mohamad Mahdi Askari and Vahid Khatibi Bardsiri** for the prediction of software program defects used artificial neural network which will better the generalization functionality of the algorithm [13]. Further support vector device approach was used together with the gaining knowledge of set of rules and evolutionary method. Thus, this led to the maximization classification margin and prevented over fitting trouble. This algorithm was examined with eleven system gaining knowledge of fashions from NASA datasets. The end drawn became that it furnished higher accuracy and precision than different models.

**Ahmet Okutan and Olcay Taner Yıldız,** proposed a new kernel technique to are expecting the number of defects in the software modules (classes or documents) [10]. The proposed method is based on a pre-computed kernel matrix which is based totally on the similarities the various modules of the software device. Novel kernel technique with present kernels within the literature (linear and RBF kernels) has been as compared and show that it achieves comparable results. Furthermore, the proposed defect prediction approach is likewise similar with a few existing well-known illness prediction methods inside the literature i.E. Linear regression and IBK. It was visible that previous to check section or preservation, developers can use the proposed technique to without problems are expecting the faultiest modules within the software program device and cognizance on them frequently as opposed to testing each and every module in the machine. This can decrease the testing attempt and the full venture fee automatically.

**N Fenton, et al.** Used Baysian networks for forecasting of reliability and defectiveness of software program [6]. It elaborates spontaneous manner factors and qualitative and quantitative degree, hence catering to the conditions of traditional software barriers.

**Jie Xu, et al.** various statistical techniques and gadget mastering strategies had been used to verify the validity of software defect prediction models [11]. Here neuro fuzzy approach turned into used. Data from ISBSG have been taken to carry out the work.

**Manu Banga,** here a brand-new computational intelligence sequential hybrid architecture involving Genetic Programming (GP) and Group Method of Data

Handling (GMDH) viz. GPGMDH had been discussed [12]. Besides GP and GMDH, a number of techniques at the ISBSG dataset has been examined. The proposed GP-GMDH and GMDH-GP hybrids outperformed all other stand-on my own and hybrid techniques. It is concluded that the GPGMDH or GMDH-GP version is the high-quality version among all different techniques for software program cost estimation.

## 4. CONCLUSIONS

The construction of models for the prediction of software bugs aims to use measures that can be relatively early in the life cycle of software development to provide reasonable initial estimates of the quality of an evolving software system. The technique of artificial intelligence is used to this, namely. ANFIS model prediction of defects in software development is a very proper technique for predicting defective in a software system module prior to implementation of the project is very important, and that a decrease in the total cost of the project and an increase in the overall success of the project rate. The prediction of defects will give one opportunity to the development team to retest modules or files for which the probability of defectiveness is high. Here showed that the developed models using ANFIS technique could be used to solve this critical problem. It is clearly demonstrated that ANFIS can be trained to accurately predict the defect of the software using variable parameters of the process. The results demonstrate that one of the real strengths of ANFIS is to work well even when training data contain noise and measurement errors. During the learning process, i.e. ANFIS are able to filter out the noise and the measurement error and actually generalize the behavior of the system.

Based on an analysis of previous results, given in the header results and discussions, is that the model of prediction of defects of software developed using ANFIS technique has been able to work well.

As a future direction, to refine the research, one may include other software and in the process metrics model it reveals the relationships among them and to determine the most appropriate ones in defect forecasting. Thus, rather than dealing with a large set of software metrics, focusing on the most effective ones will improve the success rate in defect prediction studies.

## REFERENCES

[1]."Fuzzy Logic Toolbox", MATLAB version R2012a

[2].JANG, J-S. R., "ANFIS-Adaptive-Network Based Fuzzy Inference System", IEEE Transactions on Systems, Man and Cybernatics, 23(3), pp 665-685, 1993.

[3].http://mdp.ivv.nasa.gov/mdp_glossary.html

[4].Parvinder S. Sandhu, Sunil Khullar, Satpreet Singh, Simranjit K. Bains, Manpreet Kaur, Gurvinder Singh, "A

Study on Early Prediction of Fault Proneness in Software Modules using Genetic Algorithm", World Academy of Science, Engineering and Technology, 2010, pp. 648-653.

[5].http://puretest.blogspot.com/2009/11/1.html.

[6].N. Fenton and M. Neil (2008) "Using Bayesian networks to predict software defects and reliability", Proc. IMechE Vol. 222 Part O: J. Risk and Reliability, pp 702-712.

[7].Norman Fenton, Paul Krause and Martin Neil,(1999) "A Probabilistic Model for Software Defect Prediction", For submission to IEEE Transactions in Software Engineering.

[8].http://puretest.blogspot.com/2009/11/1.html.

[9].Bibi S., Tsoumakas G., Stamelos I., Vlahavas I.(2006), "Software Defect Prediction Using Regression via Classification", IEEE International Conference on Computer Systems and Applications, pp.330 – 336.

[10]. Ahmet Okutan, Olcay Taner Yıldız,(2012) "Software defect prediction using Bayesian networks", Empir Software Eng (2014) 19:154–181 © Springer Science+Business Media, LLC.

[11].Jie Xu, [2]Danny Ho and [1]Luiz Fernando Capretz, "AN EMPIRICAL STUDY ON THE PROCEDURE TO DERIVE SOFTWARE QUALITY ESTIMATION MODELS", International journal of computer science & information Technology (IJCSIT) Vol.2, No.4, 2010.

[12].Manu Banga, "Computational Hybrids Towards Software Defect Predictions", International Journal of Scientific Engineering and Technology Volume 2 Issue 5, pp : 311-316, 2013.

[13]. Mohamad Mahdi Askari and Vahid Khatibi Bardsiri (2014), "Software Defect Prediction using a High Performance Neural Network", International Journal of Software Engineering and Its Applications Vol. 8, No. 12 (2014), pp. 177-188.

[14]. Kamaljit Kaur (2012), "ANALYSIS OF RESILIENT BACK-PROPAGATION FOR IMPROVING SOFTWARE PROCESS CONTROL" International Journal of Information Technology and Knowledge Management July-December 2012, Volume 5, No. 2, pp. 377-379.

[15]. Mrs.Agasta Adline, Ramachandran. M (2014), "Predicting the Software Fault Using the Method of Genetic Algorithm", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Special Issue 2, pp 390-398.

[16]. Xiaoxing Yang, et.al. (2014), IEEE TRANSACTIONS ON RELIABILITY, This article has been accepted for inclusion in a future issue of this journal.

**BIOGRAPHIES**

My name is Vijay Kumar Boudh I have completed B.Tech from Saroj Institute of Technology and Management, Lucknow and pursuing M.Tech in Computer Science and Engineering from Goel Institute Of Technology And Management Lucknow.

Mr. Prabhat kr. Bharti received a Master's degree and a pursuing Ph.D. in Computer Science. Currently working as Assistant Professor in the department of Computer Science and Engineering, Goel Institute of Technology and Management, Lucknow.

Mr. Deena Nath received a Master's degree in Computer Science. Currently working as an Assistant Professor in the department of Computer Science and Engineering, Goel Institute of Technology and Management, Lucknow.