International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 07 | July 2019 www.irjet.net p-ISSN: 2395-0072

A NOVEL APPROACH ON COMPUTATION INTELLIGENCE TECHNIQUE FOR SOFTWARE DEFECT PREDICTION DURING EARLY STAGE OF SOFTWARE DEVELOPMENT

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
[2,3]Assistant Professor, Department of Computer Science & Engineering, Goel Institute of Technology & Management, Lucknow

---

ABSTRACT:- Faults in software systems continue to be a major problem. Fault is a flaw that results in failure. These software defects may lead to degradation of the quality which might be the underlying cause of failure. High quality of software is ensured 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

Faults in software program structures continue to be a chief hassle [4]. Software worm is an errors, mistake, flaw, failure, or fault in a pc application that prevents it from behaving as supposed (e.g., generating an incorrect result) [5]. A software fault is a defect that reasons 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 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. APPROACHES

It has carried out a step-by-step approach for the development and successful implementation of the ANFIS approach for prediction of defects in software model. In the present work, software metrics and the set of data

used are the NASA mission-critical software projects, which are all high security and complex real time system. They are taken from the data set repository of PROMISE Software Engineering placed at the disposal of the public in order to promote models that predict repeatable, verifiable, refutable, and / or improve software engineering. They are class-level data for CM1. This includes a language attribute (defects) to indicate defectiveness. Different approaches include:

Proper selection of input parameters that have a close relationship with the model prediction errors in software development.

Use permutation and combination to divide data sets into subsets of training and testing, in such a way that results in the development of an appropriate forecasting model. These training and test data will be used during the generation of the MATLAB code. In addition, you should check the training data to have all the features of the problem in order to obtain an effective development of the model;

A. In case of ANFIS model development,

- For the limited database, in order to deduct the large computation time and deduct the number of rules, the choosing of the rule appropriate via extraction method. The subtractive clustering is performed for an effective input space partition. In the case of the subtractive cluster, is the proper choosing of the radius of the cluster by trial and error.
- The proper selection of functions parameterized, known as membership functions, is a significant step in the development of an optimal model. Gaussian membership function here is selected by delinquency for the development of prediction model.
- Selection appropriate algorithm of training, the original step size and the number of the training period.
- Acceptance and analysis of ANFIS models.

B. An analytical study of the technique followed by conclusions on the model developed on the basis of performance criteria.

3. RESULTS

ANFIS model having twenty input variables are trained and tested by ANFIS method and their performances compared and evaluated based on training and testing data. The best fit model structure is determined according to criteria of performance evaluation. The performances of the ANFIS model for checking the over fitting of the model are shown in Fig. 1 & 2 below and their RMSE values both for training and testing data are shown in Table 1 below. Fig. 3 is the graphical representation of the RMSE values for various model development stages, both for training and testing data. The low RMSE value obtained during testing phase clearly shows that the model so developed has been able to address the issue, i.e.it has performed well for predicting software defect during software development stage.

Table 1 RMSE Values for Datasets after using ANFIS

Model Development Stages	RMSE	
	Trg. Data	Chk. Data
System generated using ANFIS	1.6568e-015	2.2408e-015
Optimized Output value	4.9431e-006	9.0379e-005
Testing the overfitting	2.5722e-007	3.8947e-007

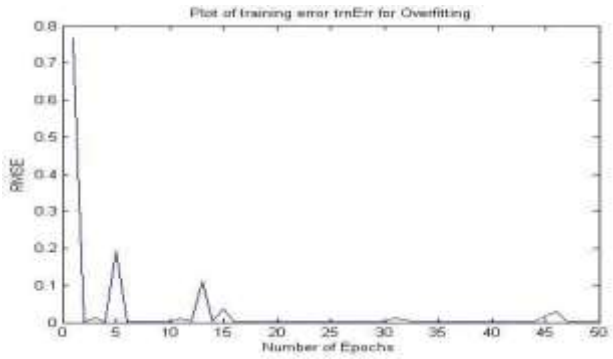

Fig 1 RMSE Plot of Training Datasets during ANFIS Training

Testing Data

Further from the analysis of the RMSE values as given in table 1 and as also from the corresponding graph given in fig. 3, both for training and testing data sets, it is clear that the ANFIS model has been able to perform better for both training and testing datasets. From the perusal of Fig. 3 it is evident that

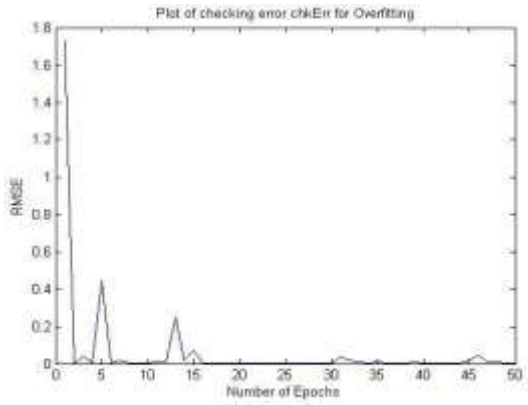

Fig. 2 RMSE Plot of Testing Datasets during ANFIS Training

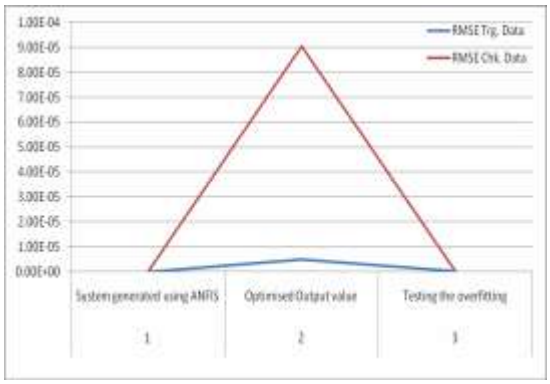

Fig. 3 Plot of RMSE Values for Training

And during the various stages of model development the RMSE values for testing datasets has shown improvement, whereas the RMSE values for training datasets has remained more or less the same. Further it is seen that during testing the over-fitting of the model, the RMSE value has improved for test data as compared to data optimization value, clearly showing that the model so developed has not over-fitted, i.e. during training phase the model has not been subjected to too much learning.

4. CONCLUSIONS

In the present chapter, applicability and capability of ANFIS techniques for Software Defect prediction has been investigated. It detects that ANFIS models are very vigorous, having quick computing power, able to tackle noisy data. The data can also be approximate. The present data that is being used in the work has the same characteristics. Due to the existence of non-linearity in the data, it is an efficient quantitative tool to forecast defect. The study has been drifting out using MATLAB simulation environment. The dataset has 22 attributes, 5 different lines of code measure, 3 McCabe metrics, 4 base Halstead measures, 8 derived Halstead measures, a branch-count, and 1 goal field as error field in linguistic form.

It is seen that the software defect prediction model developed using ANFIS technique has been able to perform well. This can be concluded from the analysis of the results given in Table 4.5 and Figure 4.6. The RMSE value obtained from ANFIS model during training stage is 1.6568e-015, whereas during testing stage it is 2.2408e-015. This again characterized the predictive capability of ANFIS technique.

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.

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.

R. 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.