

Smart Prediction Method of Software Defect Using Neuro-Fuzzy Approach

Sunil Kumar Singh and Raj Shree

Department of Information Technology, Babasaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh, India
E-Mail: jmisunil@gmail.com, rajshree.bbau2009@gmail.com

Abstract - Faults in software program structures continue to be a primary problem. A software fault is a disorder that reasons software failure in an executable product. A form of software fault predictions techniques were proposed, however none has proven to be continually correct. So, on this examine the overall performance of the Adaptive Neuro Fuzzy Inference System (ANFIS) in predicting software program defects and software program reliability has been reviewed. The datasets are taken from NASA Metrics Data Program (MDP) statistics repository. In the existing work a synthetic intelligence technique viz. Adaptive Neuro Fuzzy Inference System (ANFIS) goes for use for software disorder prediction.

Keywords: Neuro-Fuzzy Approach, Smart Prediction, Software, Defect

I. INTRODUCTION

Faults in software program software structures stay a prime trouble [1]. High first-rate of software program software is ensured with the aid of Software reliability and Software exquisite assurance. Both those principles are drawn in throughout the software improvement and preservation method. The activities like the overall performance evaluation, practical assessments, quantifying time and price range along facet dimension of metrics are used to ensure best. A software pc virus is a blunders, flaw, mistake, failure, or fault in a computer software program that forestalls it from behaving as meant (e.g., generating an incorrect result) [2]. A software fault is an illness that reasons software failure in an executable product. Most bugs rise up from mistakes and mistakes made by way of the use of people in either a software program's deliver code or its layout, and a few are resulting from compilers producing wrong code. Knowing the causes of possible defects further to identifying modern day software program method areas that may want attention from the initialization of a undertaking should store cash, time and artwork [19]. The opportunity of early estimating the capacity faultiness of software program can also need to assist on planning, controlling and executing software program application development sports activities [3].

The objective inside the creation of models of software program software blunders prediction is to use measures that may be acquired pretty early in the software program improvement life cycle to offer low priced preliminary estimates of superb of an evolving software program device [15].

In the existing paper an Adaptive Neuro Fuzzy Inference System (ANFIS) Approach might be applied for the improvement of a green predictive model using Subtractive Clustering Algorithm. For this NASA's Metrics Data Program (MDP) containing software metric records and mistakes information on the characteristic/method level has been used to validate the set of rules [11]. This one includes a numeric characteristic (NUMDEFECTS) to suggest defectiveness.

Software assignment and first-rate managers want to juggle a combination of unsure factors, consisting of gear, personnel, improvement strategies and testing techniques to reap the shipping of a first-rate product to price range and on time. Each of those uncertain elements affects the arrival, detection and correction of defects the least bit levels inside the improvement lifestyles cycle from initial requirements to product delivery. In order to gain software software first-class at some point of development specific emphasis desires to be implemented to the subsequent three sports in particular: Defect prevention; Defect detection; Defect correction. Thus as a way to deal with the above cited issues. An attempt has been made within the present paintings for software program errors prediction.

Defective software application modules motive software program application screw ups, increase development and upkeep fees, and reduce consumer pleasure. The present work strives to improve software program pleasant and testing performance by way of building predictive models from code attributes to enable a nicely timed identification of fault-prone modules. Identifying and locating defects in software application projects is a hard work [14]. Especially, at the same time as project sizes broaden, this task turns into expensive with today's checking out and evaluation mechanisms [13]. On the opposite hand, measuring software in a non-forestall and disciplined way brings many benefits which include correct estimation of task prices and schedules, and enhancing product and gadget traits. Detailed analysis of software program metric data also offers large clues approximately the locations of viable defects in a programming code [12], [20].

II. RELATED WORK

Norman Fenton *et. al.*, [4], have described a probabilistic model for software program sickness prediction. This

version can't best be used for assessing ongoing obligations, however additionally for exploring the viable effects of diverse software machine development sports. If prices may be associated with machine enhancements, and advantages assessed for the predicted improvement in software program first-rate, then the model may be used to help sound choice making for SPI (Software Process Improvement).

Ahmet Okutan, *et al.*, (2012)[5], proposed a unique approach the use of Bayesian networks to explore the relationships amongst software metrics and illness proneness. We use 9 data units from Promise information repository and display that RFC, LOC, and LOCQ are extra effective on infection proneness.

Mrinal Singh Rawat *et al.*, (2012)[6], diagnosed causative factors which in flip suggest the treatments to enhance software program first-rate and productiveness. The paper also showcases on how the several defect prediction fashions are applied ensuing in reduced significance of defects. They furnished using several gadget mastering techniques for the software program program fault prediction trouble. The unfussiness, ease in version calibration, consumer reputation and prediction accuracy of those superb estimation techniques display its sensible and applicative magnetism. These modeling structures may be used to obtain nicely timed fault predictions for software program additives presently beneath improvement, supplying precious insights into their extraordinary. The software program application exceptional assures crew can then employ the predictions to use available property for acquiring fee effective reliability enhancements.

Supreet Kaur, *et al.*, (2012)[7], studied the performance of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is evaluated for Fault prediction in Java based Object Oriented Software systems and C++ language based totally software program application components.

Xiao-dong Mu *e. al.*, (2012)[8], in their artwork to improve the accuracy of software program software disease prediction, a co evolutionary set of guidelines primarily based on the competitive business enterprise is suggest for software ailment prediction. During this set of guidelines, firstly, opposition mechanism is brought to business enterprise co evolutionary set of rules. Then, 3 evolution operators which may be reduced operator, allied operators and disturbed operators are advanced for evolution of populace. And opposition is taken into consideration for calculate the fitness characteristic. When the set of guidelines carried out into software program contamination prediction, it improves the accuracy of software program application prediction via will increase the variety of population.

N Fenton, *et al.*, (2008)[9], in their paper reviewed the usage of Bayesian networks (BNs) in predicting software program defects and software program reliability. The approach

allows analysts to contain causal technique elements further to combine qualitative and quantitative measures, as a end result overcoming a number of the general limitations of traditional software metrics strategies. Using such 'dynamic discretization' algorithms effects in extensively stepped forward accuracy for defects and reliability prediction kind models.

Jie Xu, *et al.*, (2010)[10], used Several statistical techniques collectively with device studying approach are implemented to verify the effectiveness of software program software metrics. Moreover, a neuro-fuzzy technique is adopted to enhance the accuracy of the estimation version. This procedure is performed primarily based on statistics from the ISBSG repository to offer its empirical fee.

Manu Banga, (2013) [11], In his paper, new computational intelligence sequential hybrid architectures regarding Genetic Programming (GP) and Group Method of Data Handling (GMDH) viz. GPGMDH.

III. METHODOLOGY

Adaptive neuro-fuzzy inference device is a fuzzy inference device implemented in the framework of an adaptive neural network [12]. By using a hybrid studying manner, ANFIS can construct an input-output mapping primarily based on each human-knowledge as fuzzy if-then guidelines and approximate club capabilities from the stipulated enter-output data pairs for neural network schooling. This system of growing a FIS using the framework of adaptive neural community is known as an adaptive neuro fuzzy inference gadget (ANFIS)[12][13]. There are two methods that ANFIS gaining knowledge of employs for updating membership feature parameters: 1) lower back propagation for all parameters (a steepest descent technique), and 2) a hybrid technique which include lower back propagation for the parameters associated with the input membership and least squares estimation for the parameters related to the output membership capabilities [16] - [18]. As a result, the training mistakes decreases, at the least regionally, during the learning manner. It applies the least-squares technique to perceive the consequent parameters that define the coefficients of each output equation inside the Sugeno-kind fuzzy rule base. The schooling system maintains till t he favored variety of training steps (epochs) or the favored root mean rectangular errors (RMSE) among the desired and the generated output is carried out. This examine uses a hybrid studying set of rules, to perceive premise and consequent parameters of first order Takagi-Sugeno type fuzzy gadget for predicting software program errors.

The algorithm for the work is given below:

1. Start MATLAB.
2. Load data. For this NASA's Metrics Data Program (MDP) containing software metric data and error data at the function/method level has been used.
3. Divide the data into training and testing datasets using Matlab commands.

4. Start ANFIS Editor using commands.
5. Load training data into ANFIS editor.
6. Generate Fuzzy Inference System (FIS) using Subtractive clustering algorithm.
 - a. Input Selection: Number and type of input / output membership functions.
7. ANFIS Training
 - a. Optimization method selection: Error tolerance, no. of epochs.
8. ANFIS Testing
 - a. Plot ANFIS output against Observed training and testing data.
9. Record the plot of training and testing datasets.
10. ANFIS Evaluation based on RMSE i.e. $RMSE < RMSE_{ref}$?
11. NO, goto step 5 and repeat step 6 to 10, else
12. END.

The ANFIS is a fuzzy Sugeno model placed in the framework of adaptive structures to facilitate learning and adaptation. The Sugeno fuzzy version changed into proposed via Takagi & Sugeno that allows you to formalize a systematic technique to generating fuzzy rules from an enter-output information set.

A usual fuzzy rule in a Sugeno fuzzy version has the format IF x is A and y is B THEN $z = f(x,y)$, wherein A and B are fuzzy sets within the antecedent; $z = f(x, y)$ is a crisp feature in the consequent.

Usually $f(x, y)$ is a polynomial inside the input variables x and y, however it may be any other functions which could accurately describe the output of the device inside the fuzzy place precise by the antecedent of the rule of thumb. If $f(x, y)$ is a first-order polynomial, than model is called as the first-order Sugeno fuzzy version. If f is a consistent, then it is called the zeroorder Sugeno fuzzy model, which can be regarded both as a special case of the Mamdani fuzzy inference gadget, wherein every rule's consequent is designated through a fuzzy singleton, or a unique case of Tsukamoto's fuzzy model where each rule's consequent is specific via a club feature of a step characteristic centered on the steady. Moreover, a zero order Sugeno fuzzy model is functionally equal to a radial basis feature community under positive minor constraints [5].

Considering a primary-order Sugeno fuzzy inference machine which incorporates guidelines:

- Rule 1: IF X is A_1 AND Y is B_1 ,
 THEN $f_1 = p_1 x + q_1 y + r_1$
 Rule 2: IF X is A_2 AND Y is B_2 ,
 THEN $f_2 = p_2 x + q_2 y + r_2$.

Figure illustrates graphically the fuzzy reasoning mechanism to derive an output f from a given input vector [x, y]. The firing strengths w1 and w2 are usually obtained as the product of the membership grades in the premise part, and the output f is the weighted average of each rule's output. To facilitate the learning of the Sugeno fuzzy model,

it is convenient to put the fuzzy model into framework of adaptive networks that can compute gradient vectors systematically. The resultant network architecture is ANFIS that is shown in Figure 2, where node within the same layer performs functions of the same type, as detailed below. Here circle indicates a fixed node, whereas a square indicates an adaptive node.

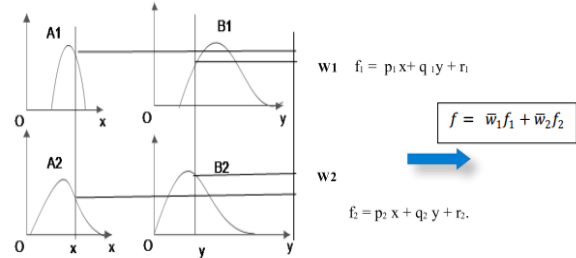


Fig. 1 First Order Sugeno Model

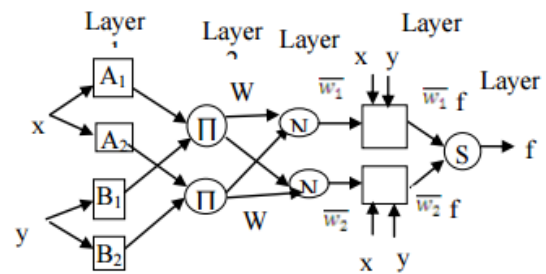


Fig. 2 ANFIS Architecture

IV. RESULT AND DISCUSSION

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 are shown in Fig. 3 & 4 and their best RMSE values based on radius of influence $r=0.75$, both for training and testing data are 0.01 and 13.25 respectively (Table I below).

TABLE I RMSE VALUES FOR DATASETS AFTER USING ANFIS

Training Data	0.014	0.0101	1.844
Testing Data	25.197	13.256	31.372
Overall Data	15.518	8.164	19.2

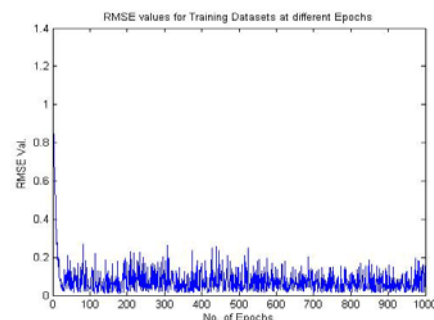


Fig. 3 RMSE Plot of Training Datasets during ANFIS Training

TABLE II SUMMERISED RESULTS OF ACTUAL AND PREDICTED DEFECT VALUES FOR TRAINING DATASETS

Act. Defect	Pre. Defect	0	-0.01251398
23	22.9890937	1	0.991823328
16	15.98918948	0	0.001242293
3	3.003743258	4	4.000252911
19	18.98897328	9	8.984964791
6	6.001008371	0	-0.00352579
3	2.99819667	0	0.000276154
3	2.996777391	8	7.999837797
3	2.997890354	101	100.9740268
4	3.990260445	0	-0.01669314
3	3.010712475	0	0.002790826
5	4.998229282	0	-0.0001285
0	-0.00010444	0	-0.00096789
4	3.997512361	0	0.006739904
0	0.024282289	0	0.000744705
0	-0.00089097	0	-0.00807808
23	22.99145436	0	-0.00575344
20	19.99665302	0	-0.01186826
0	-0.00109047	2	1.994642146
0	0.00366517	0	-0.00845106
0	-0.0012657	0	-0.00584906
2	2.001206132	0	3.76E-05
0	0.000810968	0	0.001169137
0	-0.02887525	0	-0.00052485
0	-0.00010768	0	-9.01E-05
14	13.99297704	0	-0.00510395
8	7.999988641	0	0.0044353
7	7.000460668	0	-0.01761119
22	21.99823083	0	0.010061132
5	4.996054261	0	0.031007541
0	-0.00552051	0	-0.04604462
0	0.00107111	7	7.000149208
4	3.984978301	0	-0.00170446
4	3.976680944	0	-1.43E-05
0	-0.00311922	42	41.97857439
0	-0.00032679	24	23.99989789
0	-0.00068376	17	16.99907937
0	0.012474512	8	7.999107988
0	-3.14E-05	4	4.00324772
0	0.013382224	6	5.999952047
4	3.996189869	3	3.000162951
0	0.000201082	17	16.99814297
2	1.999998524	9	9.000538086
0	0.000460854	0	0.000111185
7	7.001433515	0	0.000732842
		32	31.9974817

A comparative chart of both observed and predicted (ANFIS_Output) software defect values for training and testing data are summarised in table II.

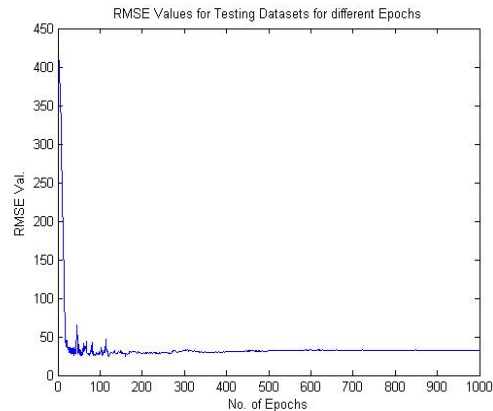


Fig. 4 RMSE Plot of Testing Datasets during ANFIS Training

V. CONCLUSION

The goal in the production of models of software blunders prediction is to use measures that can be obtained quite early inside the software program development existence cycle to provide reasonable initial estimates of pleasant of an evolving software program device. For this use of artificial intelligence technique, viz. ANFIS for the development of software program defect prediction version is a very suitable technique due to the fact predicting the faulty modules in a software device prior to challenge deployment is a totally essential hobby, since it leads to a lower inside the total price of the challenge and an growth in universal mission fulfillment fee. Defect prediction will deliver one more chance to the development crew to retest the modules or files for which the defectiveness possibility is excessive.

REFERENCES

- [1] Parvinder S. Sandhu, Sunil Khullar, Satpreet Singh, Simranjit K. Bains, Manpreet Kaur and 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.
- [2] S. Bibi, G. Tsoumakas, I. Stamelos and I. Vlahavas, "Software Defect Prediction Using Regression via Classification", *IEEE International Conference on Computer Systems and Applications*, March 8, pp. 330 - 336, 2006.
- [3] Norman Fenton, Paul Krause and Martin Neil, "A Probabilistic Model for Software Defect Prediction", *IEEE Transactions in Software Engineering*.
- [4] Ahmet Okutan and Olcay Taner Yıldız, "Software defect prediction using Bayesian networks", *Empir Software Eng* (2014) 19:154–181 © Springer Science+Business Media, LLC, 2012.
- [5] Mrinal Singh Rawat and Sanjay Kumar Dubey, "Software Defect Prediction Models for Quality Improvement: A Literature Study", *IJCSI International Journal of Computer Science Issues*, Vol. 9, No. 2, Issue 5, 2012, pp. 288-296.
- [6] Supreet Kaur and Dinesh Kumar, "Software Fault Prediction in Object Oriented Software Systems Using Density Based Clustering Approach", *International Journal of Research in Engineering and Technology (IJRET)*, Vol. 1, No. 2 March, 2012, ISSN: 2277-4378.

- [7] Xiao-dong Mu, Rui-hua Chang and Li Zhang, "Software Defect Prediction Based on Competitive Organization CoEvolutionary Algorithm", *Journal of Convergence Information Technology (JCIT)*, Vol. 7, No. 5, 2012.
- [8] N Fenton, M Neil and D Marquez, "Using Bayesian networks to predict software defects and reliability", Review Paper, *Proc. IMechE*, Vol. 222, Part O: J. Risk and Reliability, 2008.
- [9] Jie Xu, Danny Ho and Luiz Fernando Capretz, "An Empirical Study on the Procedure to Derive Software Quality Estimation Models", *International Journal of Computer Science and Information Technology (IJCSIT)*, Vol. 2, No. 4, 2010.
- [10] Manu Banga, "Computational Hybrids towards Software Defect Predictions", *International Journal of Scientific Engineering and Technology*, Vol. 2, No. 5, pp. 311-316, 2013.
- [11] J. S. R. Jang, "ANFIS-Adaptive-Network Based Fuzzy Inference System", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 3, pp. 665-685, 1993.
- [12] Prabhishek Singh and Raj Shree, "Quantitative Dual Nature Analysis of Mean Square Error in SAR Image Despeckling", *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 9, No. 11, pp. 619-622, Nov. 2017.
- [13] Prabhishek Singh, Raj Shree, "A Comparative Study to Noise Models and Image Restoration Techniques", *International Journal of Computer Applications (0975 – 8887)*, Vol. 149, No. 1, September 2016.
- [14] Prabhishek Singh, Raj Shree, "Statistical Quality Analysis of Wavelet Based SAR Images in Despeckling Process", *Asian Journal of Electrical Sciences (AJES)*, Vol. 6, No. 2, pp. 1-18, July-December 2017.
- [15] Prabhishek Singh and Raj Shree, "A new homomorphic and method noise thresholding based despeckling of SAR image using anisotropic diffusion", *Journal of King Saud University – Computer and Information Sciences*, 2017, <http://dx.doi.org/10.1016/j.jksuci.2017.06.006>
- [16] Prabhishek Singh, Raj Shree and Manoj Diwakar, "A new SAR image despeckling using correlation based fusion and method noise thresholding", *Journal of King Saud University – Computer and Information Sciences*, 2018, <https://doi.org/10.1016/j.jksuci.2018.03.009>
- [17] Prabhishek Singh and Raj Shree, "Statistical Modelling of Log Transformed Speckled Image", *International Journal of Computer Science and Information Security*, Vol. 14, No. 8, pp. 426-431, August 2016.
- [18] Prabhishek Singh and Raj Shree, "A New Computationally Improved Homomorphic Despeckling Technique of SAR Images", *International Journal of Advanced Research in Computer Science*, Vol. 8, No. 3, March April 2017.
- [19] Prabhishek Singh and Raj Shree, "Analysis and effects of speckle noise in SAR images", *IEEE International Conference on Advances in Computing, Communication, Automation (ICACCA)*, 2016.
- [20] Prabhishek Singh and Raj Shree, "A new SAR image despeckling using directional smoothing filter and method noise thresholding", *Eng. Sci. Tech., Int. J.*, 2018, <https://doi.org/10.1016/j.jestch.2018.05.009>.
- [21] Kavita Sahu, R Shree, R Kumar, "Risk Management Perspective in SDLC", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, No. 3, 2014.