International Conference on Recent Advancement in Science & Technology- 2020
(ICRAST-2020)

Development of Software Defect Prediction Model using Artificial Intelligence Method

Ritu Vishwkarma[1], Madan Kushwaha[2]
Dept of Computer Science & Engineering
Bansal Institute of Engineering and Technology, India,
er.reetuvish@gmail.com, kushwaha.86@gmail.com

Abstract: Software illness prediction paintings focuses on the variety of defects final in a software program system. A prediction of the variety of closing defects may be used for selection making. An accurate prediction of the number of defects in a software program product at some point of gadget testing contributes now not only to the management of the system trying out process however additionally to the estimation of the product's required protection. In the existing paper an Adaptive Neuro Fuzzy Inference System (ANFIS) Approach has been applied for the improvement of an green predictive model using Substractive Clustering Algorithm. For this NASA's Metrics Data Program (MDP) containing software metric information and error information on the function/technique stage has been used to validate the algorithm. The experimental consequences display that the proposed algorithm is powerful for software illness prediction.

***Keywords*- Software Defect, RMSE, ANFIS, MDP**

1. INTRODUCTION:

Software metrics-based excellent prediction fashions may be powerful tool for identifying the wide variety of defects of the modules. The use of such models previous to every launch making plans or re-making plans of the gadget, or maybe deployment of which can drastically improve machine exceptional. A disorder prediction model is calibrated the use of metrics from a past release or similar task, and is then implemented to modules presently underneath improvement. Subsequently, a well timed prediction of which modules wishes more attempt to do away with the defects, can be obtained. Over the beyond decades years, several empirical research have been executed to are expecting the fault proneness fashions. Software illness prediction research can be classified as statistical and system getting to know (ML) tactics. And using machine studying tactics to fault prediction modeling is more famous[7]. Unfortunately, this problems of software program illness prediction have not resolved thoroughly. And not one of the techniques have accomplished big applicability within the software industry due to several motives, together with the challenge of testing useful resource, the lack of software tools to automate this software illness prediction, the unwillingness to accumulate the software program disorder facts, many strategies based on the personal software program information, and the alternative sensible problems [6]. Adaptive Neuro Fuzzy Inference System (ANFIS) proposed by way of R. Jang,[10] is an evolutionary artificial intelligence approach[10] and has been applied into many regions including software program disorder prediction. It has the gain of allowing the extraction of fuzzy regulations from numerical facts or expert understanding and adaptively constructs a rule base. Moreover, it may adapt the complicated conversion of human intelligence to fuzzy systems. In the existing work an ANFIS technique using subtractive clustering set of rules has been used implemented to resolve the trouble of software defect prediction.

The rest of the paper is prepared as follows: Section II describes the Literature Review. Section III deals with information used, IV the technique a part of paintings carried out, accompanied by way of consequences and discussions in segment V. In the final section, on the premise of the discussion various conclusions are drawn and the future scope for the existing work is mentioned.

2. Literature Survey:

Various techniques, which includes linear regression, discriminate evaluation, choice timber, neural networks and so on. Were developed and implemented to expect defects in software program. Munson et al. [1] look into linear regression models and discriminate evaluation to finish the overall performance of the latter is better. Catal et al [2] evolved an Eclipse-primarily based software fault prediction equipment for Java programs, and naïve bays selected as the plug-in for the gear. Norman Fenton et al.[3] used dynamic Bayesian nets for predicting software defects. Rather than relying only on facts from previous variations, his method uses causal fashions of the Project Manager's understanding and protecting mechanisms. Bullard et al. [4] employ a rule-primarily based class version in a telecommunication gadget and stated that their version produces lower false positives, which might be considered as excessive value classification mistakes.. Ahmet Okutan, et. Al., (2012)[6] used Bayesian networks to decide the probabilistic influential relationships among software metrics and disorder proneness. Mrinal Singh Rawat, et. Al.,(2012)[7] identified causative elements which in flip advise the treatments to enhance software program satisfactory and productivity. It additionally confirmed on how the diverse disorder prediction models are applied resulting in decreased importance of defects. Supreet Kaur, et. Al., (2012)[8] confirmed that the overall

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 software program components. Xiao-dong Mu, et. Al.,(2012)[9] In order to improve the accuracy of software program defect prediction, a coevolutionary set of rules based totally on the competitive corporation is recommend for software program illness prediction. Experiment based on the 5 datasets from NASA is used to validate the method. The experimental outcomes display that the proposed method is powerful.

3. DATA USED:

The software metrics and dataset used in this study are mission critical NASA software projects [5], which are all high assurance and complex real-time system. They are taken from PROMISE Software Engineering Repository data set made publicly available in order to encourage repeatable, verifiable, refutable, and/or improvable predictive models of software engineering. They are Class-level data for KC1. This one includes a numeric attribute (NUMDEFECTS) to indicate defectiveness. The descriptions of the features are taken from http://mdp.ivv.nasa.gov/mdp_glossary.html.

Table 1. Input and Output Variables for ANFIS Model.

Input Variable	LOC_BLANK BRANCH_COUNT LOC_CODE_AND_COMMENT LOC_COMMENTS CYCLOMATIC_COMPLEXITY DESIGN_COMPLEXITY ESSENTIAL_COMPLEXITY LOC_EXECUTABLE HALSTEAD_CONTENT HALSTEAD_DIFFICULTY HALSTEAD_EFFORT HALSTEAD_ERROR_EST HALSTEAD_LENGTH HALSTEAD_LEVEL HALSTEAD_PROG_TIME HALSTEAD_VOLUME NUM_OPERANDS NUM_OPERATORS NUM_UNIQUE_OPERANDS NUM_UNIQUE_OPERATORS LOC_TOTAL
Output Variable	NUMDEFECTS

4. ANFIS MODEL DEVELOPMENT

4.1 Parameter Selection

ANFIS [11],[14] is a sensible integration of FIS and ANN, able to learning, excessive-stage wondering and reasoning and it combines the advantages of those two techniques right into a single tablet. Identification of the rule base is the key of a FIS. The troubles are (1) there are not any widespread techniques for reworking human information or experience into rule base; and (2) it's far required to in addition tune the MFs to minimise the output error and to maximize the performances. Thus whilst generating a FIS the usage of ANFIS, it's miles crucial to select right parameters, which includes the number of club features (MFs) for each character antecedent variables. It is also essential to pick right parameters for studying and refining system, which includes the preliminary step size (ss). In the existing paintings the commonly used rule extraction approach carried out for FIS identity and refinement is subtractive clustering. The ANFIS is simulated the usage of the MATLAB Fuzzy Logic Toolbox [12].

Here the preliminary parameters of the ANFIS are diagnosed using the subtractive clustering technique [13]. However, the parameters of the subtractive clustering algorithm nevertheless need to be certain. The clustering radius is the maximum essential parameter in the subtractive clustering algorithm and is optimally determined thru an ordeal and errors manner. By varying the clustering radius ra between 0.1 and 1 with a step size of zero.01, the surest parameters are sought by minimizing the root mean squared blunders obtained on a consultant validation set. Clustering radius rb is chosen as 1.5 ra. Default values are used for different parameters inside the subtractive clustering set of rules.

Gaussian membership features are used for every fuzzy set inside the fuzzy gadget. The number of membership capabilities and fuzzy regulations required for a specific ANFIS is decided thru the subtractive clustering algorithm. Parameters of the Gaussian membership function are optimally decided the use of the hybrid mastering set of rules. Each ANFIS is trained for 1000 epochs.

Gaussian club feature has been used because the input club feature and linear membership characteristic for the output characteristic. Here separate sets of input and output records has been used as enter arguments. In MATLAB genfis2 generates a Sugeno-type FIS structure the usage of subtractive clustering. Since there may be most effective one output, genfis2 has been used to generate an preliminary FIS for ANFIS schooling. Genfis2 accomplishes this through extracting a set of guidelines that models the statistics behaviour. The rule extraction approach first uses the subclust characteristic to decide the number of policies and antecedent club features after which uses linear least squares estimation to decide every rule's consequent equations. This feature returns a FIS structure that carries a fixed of fuzzy policies to cover the feature area.

Table 1 shows the input and output version parameters used for version improvement. The parameters used inside the version for training ANFIS are given in Table 2 and the guideline extraction method used are given in Table 3. Table four summerizes the effects of sorts and values of model parameters used for training ANFIS

Table 2. Parameters used in all the models for training ANFIS

Rule extraction method used	Subtractive clustering
Input MF type	Gaussian membership (‘gaussmf’)
Input partitioning	variable
Output MF Type	Linear
Number of output MFs	one
Training algorithm	Hybrid learning
Training epoch number	1000
Initial step size	0.01

Table 3. Rule extraction method used for training ANFIS

Rule Extraction Method	Type
And method	'prod'
Or method	'probor'
Defuzzy method	'wtever'
Implication method	'prod'
Aggregation method	'max'

Table 4. Values of parameters used for training ANFIS

No. of nodes	244
No. of linear parameters	110
No. of non-linear parameters	210
Total no. of parameters	320
No. of training data pairs	90
No. of testing data pairs	55
No. of fuzzy rules	5

5. RESULTS AND DISCUSSIONS

Here the ANFIS model has been educated tested through ANFIS method and their overall performance for the great prediction version are evaluated and compared for education and trying out records units separately. The fine RMSE performances of the ANFIS model (for cluter radius r=0.Seventy five) both for education and checking out datasets were plotted one at a time in Fig. 1 & Fig.2 and their corresponding RMSE values for schooling and trying out datasets for distinctive clustering radius are summarized in Table 5.

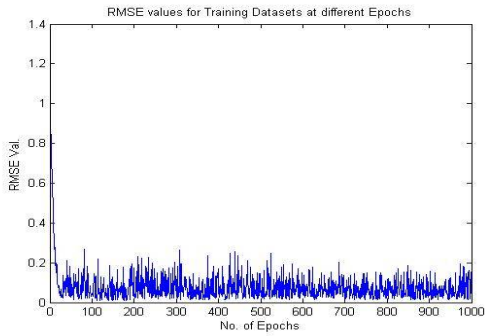

Fig. 1 Graphical plot of RMSE value variation during training

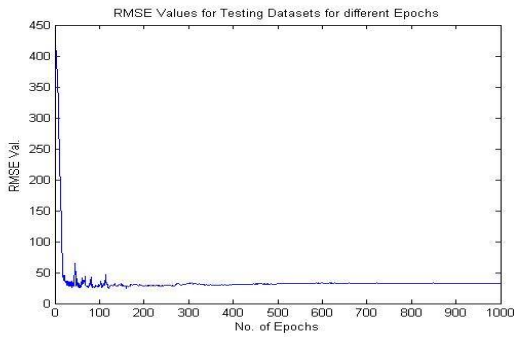

Fig. 2 Graphical plot of RMSE value variation during testing

Table 5. RMSE values for different Clustering radius

RMSE for diff. radius of influence (r)			
	r=0.5	**r=0.75**	**r=1.0**
Trg. Data	0.014	0.0101	1.844
Tst. Data	25.197	13.256	31.372
Overall Data	15.518	8.164	19.2

From the perusal of the facts given in tables 5 it's miles inferred that for 3 specific clustering radius zero.Five, 0.Seventy five and 1.Zero, the prediction model carried out pleasant for r=zero.75, having RMSE value of 0.01 and 13.256 for for schooling and trying out segment. The equal has been plotted in Fig. 1 & 2, whose analysis reveals that in training segment (Fig.1), there may be a sudden fall inside the RMSE cost during the primary 20 epochs, and then there may be more or much less gradual trade in RMSE values, having a minimal value of 0.0101 and a most cost of 1.0267. On the alternative hand, at some stage in testing segment (Fig.2) of ANFIS schooling first of all there's sudden fall in RMSE value, and then there is greater or less regular cost upto epochs 1000. Thus there is a minimum and most RMSE cost of 24.66 and 415.29. Also from the perusal of the data given in Table five it is able to be inferred that ANFIS has accomplished higher for the duration of schooling section than checking out phase however its typical RMSE price is 8.164.

Thus, it is clean that right selection of influential radius which affects the cluster outcomes without delay in ANFIS the usage of substractive clustering rule extraction approach , has ended in reduction of RMSE each for schooling and testing information units. Hence, it's miles seen that for small size schooling data, ANFIS has performed well.

In order to depict how nicely ANFIS has executed, a comparative plot of real disorder as opposed to expected attempt, the usage of ANFIS approach, has been proven in Fig. 5 & 6, the usage of facts given in Table 6 & 7. From the graph it is seen that ANFIS version line almost closely follows the actual disorder line. This again depicts the superiority of ANFIS approach for disorder prediction.

Table 6. Summerised Results of Sctual and Predicted Defect Values for Training Datasets

Act. Defect	Pre. Defect		
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

Table 7:- Summerised Results of Sctual and Predicted Defect Values for Testing Datasets

Act. Defect	Pre. Defect		
20	18.31263759	0	-0.01856829
14	-3.64604818	0	-0.03721321
3	-6.02204598	26	2.284127638
2	7.487105594	4	-13.2145005
0	-12.6343929	0	-16.0773878
6	29.53264229	0	3.530103455
19	28.79311684	6	-13.5319316
13	31.89483623	0	3.075540284
0	-16.4369616	11	38.17036328
0	-45.9344396	11	24.14581483
0	0.074209395	3	3.409076182
0	-5.12362697	0	0.117551486

6	13.86958413	13	2.415202415
0	1.542898	0	1.470573004
2	0.318371452	0	0.49834135
0	1.070233258	0	15.72977697
1	-0.8881197	0	0.051507761
0	24.6483655	0	0.134772179
0	-0.398183	0	-19.1858159
0	0.072438033	0	0.009541944
0	16.48991558	0	0.374123026
0	15.39989893	0	-0.27895808
0	-1.31037766	0	-0.02811841
0	-2.06176251	0	0.00491892
0	23.14399417	0	-0.02811841
1	24.10270153	0	3.276273884
		0	-0.02811841
		0	3.276273884

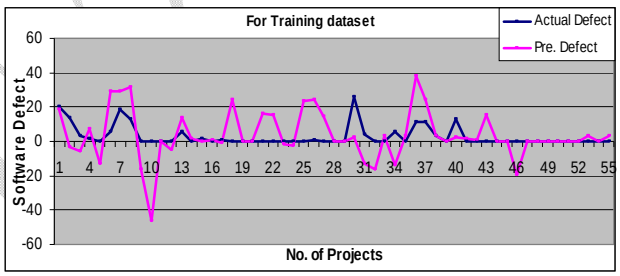

Fig 5 Comparative plot of Actual and Predicted Software Defect for Training Datasets

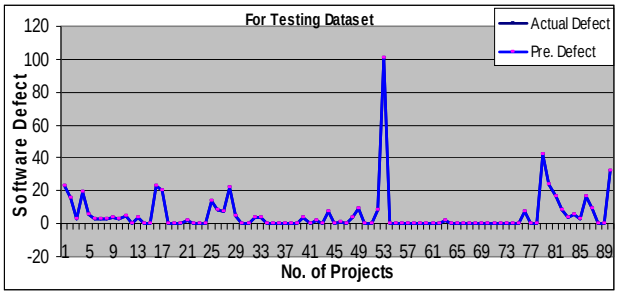

Fig 6 Comparative plot of Actual and Predicted Software Defect for Testing Datasets

Finally, Figure 7, shows the scatter plot of Actual defect versus Estimated defect using ANFIS. The figures wisely demonstrate that (1) the model performance is in general accurate in case of ANFIS, where all data points roughly fall onto the line of agreement; (2) model using ANFIS is consistently superior for software defect prediction.

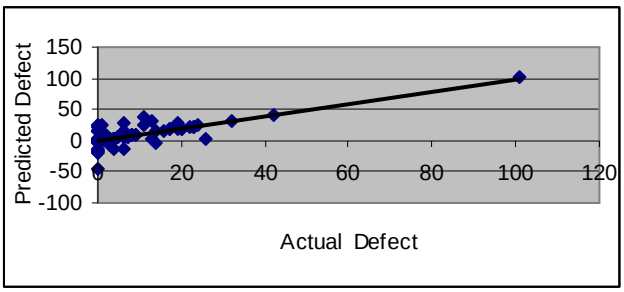

Fig 7 Scatter Plot of Actual Vs. Predicted Defect using ANFIS

6. CONCLUSION:

In the present paper, applicability and capability of ANFIS techniques for software illness prediction has been investigated. It is seen that ANFIS models are very robust, characterised with the aid of speedy computation, capable of managing the noisy and approximate statistics which can be common of records used here for the present have a look at. Due to the presence of non-linearity inside the records, it's far an green quantitative device to predict attempt estimation. The studies has been accomplished the usage of MATLAB simulation environment. In all twenty input variable were used, along with numerous software program metrics and one output variable as software program illness.

Here the preliminary parameters of the ANFIS are diagnosed using the subtractive clustering method. Gaussian club functions (given in earlier segment) are used for each fuzzy set inside the fuzzy device. The quantity of club functions and fuzzy regulations required for a selected ANFIS is decided through the subtractive clustering algorithm. Parameters of the Gaussian membership function are optimally decided using the hybrid mastering algorithm. Each ANFIS has been educated for one thousand epochs.

From the evaluation of the above consequences, given below heading Results and Discussions, it is seen that the software defect prediction version evolved the use of ANFIS technique has been capable of carry out properly. This may be concluded from the analysis of the results given in Table five. The RMSE cost received from ANFIS model for r=zero.75 is 8.164, that is lower than the ones for r=zero.Five & 1.Zero. Further from Fig. Five, & 6 and Table 7 it's far seen that ANFIS version line nearly carefully follows the real defect line. This once more depicts the superiority of ANFIS approach as a predictor tool.

REFERENCES

[1] John C. Munson and Taghi M. Khoshgoftaar, "The Detection of Fault-Prone Programs" , IEEE Transactions on Software Engineering, vol. 18, pp. 423-433, 1992. 13

[2] Cagatay Catal, Ugur Sevim and Banu Diri, "Practical development of an Eclipse-based software fault prediction tool using naïve bayes algorithm", Expert System with application, vol. 38, no. 3, pp. 2347~2353, 2011. 16

[3] Norman Fenton, Martin Neil, William Marsh, Peter hearty, David Marquez, Paul Krause and Rajat Mishra, "Predicting software defect in varying development lifecycles using Bayesian nets", Information and Software Technology, vol. 49, pp. 32~43,2007. 18

[4] Lofton A. Bullard, Taghi M. Khoshgoftaar and Kehan Gao, "An application of a rule-based model in software quality classification" , In Proceeding of Sixth International Conference on Machine Learning and Applications, pp. 204 ~210, 2007. 21

NASA metrics data program, http://mdp.ivv.nasa.gov/mdp_glossary.html. 2012. 22

[5] Ahmet Okutan, et. al., (2012), "Software defect prediction using Bayesian networks", Empir Software Eng (2014) 19:154–181

[6] Mrinal Singh Rawat, et. al.,(2012), "Software Defect Prediction Models for Quality Improvement: A Literature Study", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2,

[7] Supreet Kaur, et. al., (2012) "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, pp 111-117.

[8] Xiao-dong Mu, et. al.,(2012), "Software Defect Prediction Based on Competitive Organization CoEvolutionary Algorithm", Journal of Convergence Information Technology(JCIT) Volume7, Number5

[9] Hammouda, K. A., *"Comparative Study of Data Clustering Techniques"*.

[10] Jang, J-S. R., (1992), *"Neuro-Fuzzy Modeling: Architecture, Analyses and Applications"*, P.hd. Thesis.

[11] Fuller, R., (1995), *"Neural Fuzzy Systems"*, ISBN 951-650-624-0, ISSN 0358-5654.17

[12] Chen, D.W. And Zhang, J.P., (2005), *"Time series prediction based on ensemble ANFIS"*, Proceedings of the fourth International Conference on Machine Learning and Cybernatics, IEEE, pp 3552-3556.10

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

[14] Anurag, R. Sharma, " Load Forecasting by using ANFIS", International Journal of Research and Development in Applied Science and Engineering, Volume 20, Issue 1, 2020.

[15] R. Sharma, Anurag, " Load Forecasting using ANFIS A Review", International Journal of Research and Development in Applied Science and Engineering, Volume 20, Issue 1, 2020.

[16] R. Sharma, Anurag, " Detect Skin Defects by Modern Image Segmentation Approach, Volume 20, Issue 1, 2020.

[17] Anurag, R. Sharma, " Modern Trends on Image Segmentation for Data Analysis- A Review", International Journal of Research and Development in Applied Science and Engineering, Volume 20, Issue 1, 2020.

[18] Young Soo Jang et. al., "Development of the cost-effective, miniaturized vein imaging system with enhanced noise reduction", International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.6, November – December 2019.

[19] Irma T. Plata1, et. al., "Development and Testing of Embedded System for Smart Detection and Recognition of Witches' Broom Disease on Cassava Plants using Enhanced Viola-Jones and Template Matching Algorithm", International Journal of Advanced Trends in Computer Science and

Engineering, Volume 8, No.6, Volume 8, No.5, September - October 2019.