Improved Approach for Software Defect Prediction using Artificial Neural Networks

Tanvi Sethi[1], Gagandeep[2]

[1,2]*Department of Computer Science, Punjabi University, Patiala, India*
[1]*tanu.2098@gmail.com,* [2]*gdeep.pbi@gmail.com*

***Abstract*: Software defect prediction (SDP) is a most dynamic research area in software engineering. SDP is a process used to predict the deformities in the software. To identifying the defects before the arrival of item or aimed the software improvement, to make software dependable, defect prediction model is utilized. It is always desirable to predict the defects at early stages of life cycle. Hence to predict the defects before testing the SDP is done at end of each phase of SDLC. It helps to reduce the cost as well as time. To produce high quality software, the artificial neural network approach is applied to predict the defect. Nine metrics are applied to the multiple phases of SDLC and twenty genuine software projects are used. The software project data were collected from a team of organization and their responses were recorded in linguistic terms. For assessment of model the mean magnitude of relative error (MMRE) and balanced mean magnitude of relative error (BMMRE) measures are used. In this research work, the implementation of neural network based software defect prediction is compared with the results of fuzzy logic basic approach. In the proposed approach, it is found that the neural network based training model is providing better and effective results on multiple parameters.**

***Keywords*: Defect, Software Defect Prediction, Software Metrics, Machine Learning technique(MLT), Fuzzy logic, Artificial Neural Network(ANN)**

I. INTRODUCTION

A software deformity is characterized as a fault, bug, failure or error in a system or a project. A software imperfection may prompts wrong, surprising yield or performs in an undesired task. Defects can be brought about by the developer's slip-ups in configuration and source code, working system that these source code and off base code delivered by compilers. Alongside unwavering quality, another issue is software quality. Amid the improvement of software one of the principle concerns is software quality. The nature of software is to release the event of imperfections in it. Deformities are pretty much yet the results of these imperfections are unsafe and may prompt increment in expense.

SDP is a procedure used to foresee the imperfections that makes a software module dependable. SDP is utilized by various associations like Blackberry, IBM, and Google etc. A SDP model includes an accumulation of software measurements. The Software measurement is characterized as an approach to quantify the software quality, particular of software and attributes of software module. The machine learning methods are utilized to fabricate a SDP Model. MLT is information driven strategy. The principle objective of ML is to develop a calculation of reasonable worth and calculations ought to be well-organized. The ML can be ordered into directed (broadly utilized), unsupervised, semi-supervised. Amid the improvement of the software, the SDLC is utilized. Subsequently, it is a decent practice to pick SDP at early conditions of SDLC. Phases of SDLC are prerequisite, outlining, coding and testing. Nine measurements refers to [16] has been taken for every one of the stages with twenty genuine software venture dataset and these measurements are spoken to in etymological term (VL, L, M, H and VH). The proposed work is based on ANN refers to the high level meta-heuristic approach. The training model in ANN is providing more accuracy on multiple parameters as compared to fuzzy logic approach

II. RELATED RESEARCH

Software Defect Prediction is a broadly used in software engineering. To enhance the nature of software and to pick up consumer loyalty the software should be defect free. The analysts and the academicians perform different exploration works. A few papers have been examined amid the period the brief of these papers are portrayed.

Kener [2] define the metric estimation is the primary part of the examination. A metric is likewise an estimation capacity. Nitty gritty study is required to think about what and how metric play out the required undertaking.

Fenton[10] proposed the causal model, likewise called as Bayesian Net which is made of two sections first is coordinated non-cyclic diagram and second likelihood disseminations. Dataset from thirty one software genuine undertakings was assembled by utilizing a survey.

Panday [1] defines the types of software module are fault inclined and non-fault inclined, these two kind of software characterizes the nature of software. Additional testing has been required for faulty module and less or no testing for non-faulty module. The ID3 calculation is utilized to arrange the software into these two sorts.

Paramshetti[12] explain the SDP is utilized as a part of software system to enhance the quality and cut the expense and time. MLT is utilized as a part of SDP, which gain from past experience and perception. It likewise disposes of the incorrect

and pointless information from dataset. Nam[5] adds the software metric like code and process metric.

Yadav[4] have explained the three prerequisite measurements are utilized and these measurements are the contribution to the FIS. Requirement Stability characterizes the consistency of the need. Experience of requirement team metric processes the applicable learning, practice and abilities of colleagues for executing the venture. Requirement default density metric figures the extent of necessities. The size metric is utilized amid improvement of software.

Rawat [8] describe the objective of the study is to improve the nature of software. For this investigation, surrendered information is arranged into expectation and grouping model. These two models anticipate the imperfections in future. Deformities are followed by SDP, which is a procedure and couple of methods like neural system, bayesian net etc.

Sandhu[11] presented measure of imperfections present in software decides the nature of the software. In this paper, neural system methodology is utilized to find whether quantitative and subjective variables can be utilized to decide the level or measure of number of faulty software module. Dataset has been taken from 31 software ventures by social occasion utilizing a poll and yield created in semantic terms.

Singh[6] define the Levenberg-Marquardt calculation is utilized for foreseeing the deformity at early phases of SDLC in view of Neural system device, which takes contribution to number. The system of one information layer, three concealed layer and one yield layer is framed which comprise of thirteen neurons. The dataset is Ant 1.7 dataset is taken from PROMISE archive and the CK and OO metric is utilized as a part of the dataset.

Saduf[13] highlight the back propagation neural network. It comprise of least one info layer, one concealed layer for middle of the road handling and one yield layer. Error is registered by subtracting genuine yield from target yield and these errors are passed back to the system.

Nam[5] present the SDP procedure was concentrated on taking into account MLT. The expectation model is grouped into two class first is twofold order and second is relapse (characterize the quantity of bugs).

Kutlubay. find and discover the deformities in the early stages in the lifecyle brings about ease for this analyst had depicted the idea of machine learning for a software module concerning their metric information and draws the confinement of machine learning in SDP is absence of data.

Prasad [7] explain the principle of the work is to improve the software quality which is characterized as the software procures a few elements like security, unwavering quality, sufficient size and practicality. The procedure used to improve the quality is information mining.

Vashisht[14] provide the framework for SDP using Neural Network. Graphical User Interface framework is used for all the phases of life cycle for predicting the defects and this framework determines the probability of defects. To train the system, this historical data of 45 projects is used.

Inayathulla[9] try to reduce the testing efforts by TERA approach. This approach uses the MLT in order to decrease the testing effort. Several models are proposed to predict the number of faults and suitable efforts are applied on each module. This is based on the MLT and testing efforts are applied to the FP modules.

Mundada[3] explain the prediction of faulty module is necessary to enhance the quality and to make software reliable. ANN is used to find the faulty module and prediction of these erroneous modules. ANN has been trained by using resilient back propagation and produced results are better than conventional back propagation algorithm.

III. EXPERIMENTAL SETUP

In proposed work, a unique model is developed and training is done using ANN with comparison to fuzzy mathematical model so that the deep investigation of defects can be done. Artificial Neural Networks (ANN) refers to the high level meta-heuristic approach to achieve the excellent efficiency in training the datasets in multiple domains.

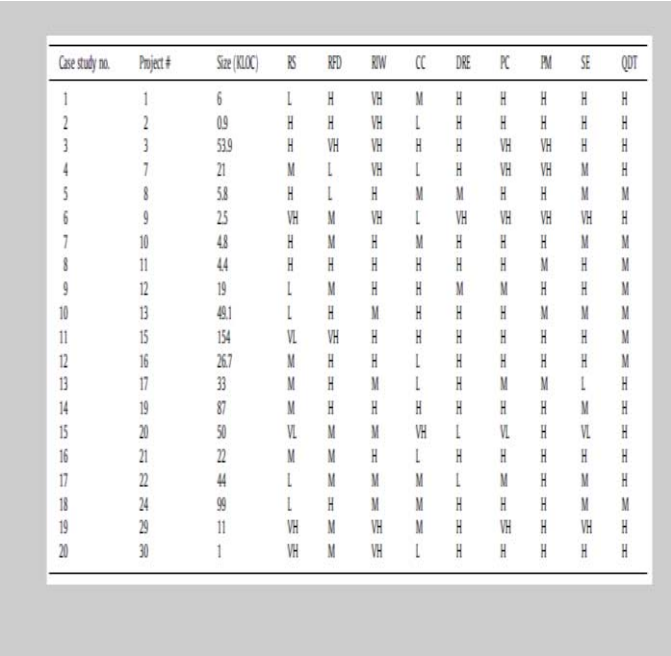
- Data Set Formation from various Sources
- Implementation of dataset using fuzzy logic in classical approach
- Features Extraction Identification of Key Aspects
- Development of a unique network for training
- Training of ANN Model
- Defects Analysis with the inherent parameters
- Deep Investigation and Predictive Analysis

IV. IMPLEMENTATION RESULTS

The software defect prediction is one of the prominent domains in software engineering which is required to be processed using new and effective algorithms. The classical approaches related to fuzzy modelling can be improved using proposed artificial neural networks.

Figure1 depicts the dataset taken from [16]. Nine measurements have been taken for every one of the stages with twenty genuine software venture dataset. Figure2 depict the rule viewer of RPDDI, takes input measurements are requirement stability (RS), requirement fault density (RFD), and requirement inspection and walkthrough (RIW). Figure3 depict the rule viewer of DPDDI takes measurement cylcomatic complexity (CC), design review effectiveness (DRE) and RPDDI. Figure4 depict the rule viewer of CPDDI takes input measurements are programmer capability (PC), process maturity (PM) and DPDDI. Figure5 depict the rule

viewer of TPDDI and staff experience (SE), quality of documented test cases (QDT) and CPDDI are measurements. Figure6 depicts the defect found by fuzzy modelling and figure7 shows the performance evaluation of fuzzy modelling based on MMRE and BMMRE measure.



Case study no.	Project #	Size (KLOC)	RS	RFD	RIW	CC	DRE	PC	PM	SE	QDT
1	1	6	L	H	VH	M	H	H	H	H	H
2	2	0.9	H	H	VH	L	H	H	H	H	H
3	3	53.9	H	VH	VH	H	H	VH	VH	H	H
4	7	21	M	L	VH	L	H	VH	VH	M	H
5	8	5.8	H	L	H	M	M	H	H	M	M
6	9	2.5	VH	M	VH	L	VH	VH	VH	VH	H
7	10	4.8	H	M	H	M	H	H	H	M	M
8	11	4.4	H	H	H	H	H	H	M	H	M
9	12	19	L	M	H	H	M	M	H	H	M
10	13	48.1	L	H	M	H	H	H	M	M	M
11	15	154	VL	VH	H	H	H	H	H	H	M
12	16	26.7	M	H	H	L	H	H	H	H	M
13	17	33	M	H	M	L	H	M	M	L	H
14	19	87	M	H	H	H	H	H	H	M	H
15	20	50	VL	M	M	VH	L	VL	H	VL	H
16	21	22	M	M	H	L	H	H	H	H	H
17	22	44	L	M	M	M	L	M	H	M	H
18	24	99	L	H	M	M	H	H	H	M	M
19	29	11	VH	M	VH	M	H	VH	H	VH	H
20	30	1	VH	M	VH	L	H	H	H	H	H

Fig. 1. DataSet of the Existing Work

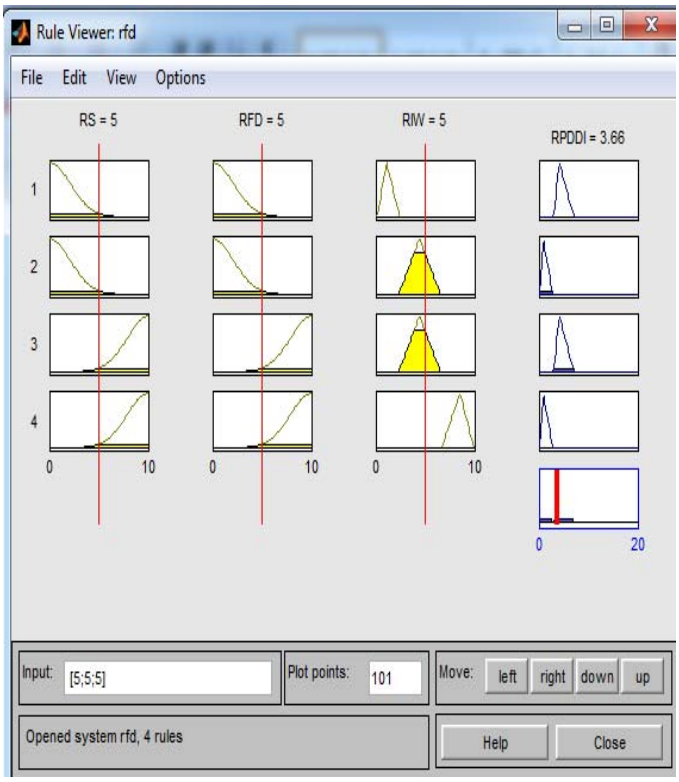


Fig. 2. Fuzzy Rule Viewer for Requirement analysis phase defect density indicator

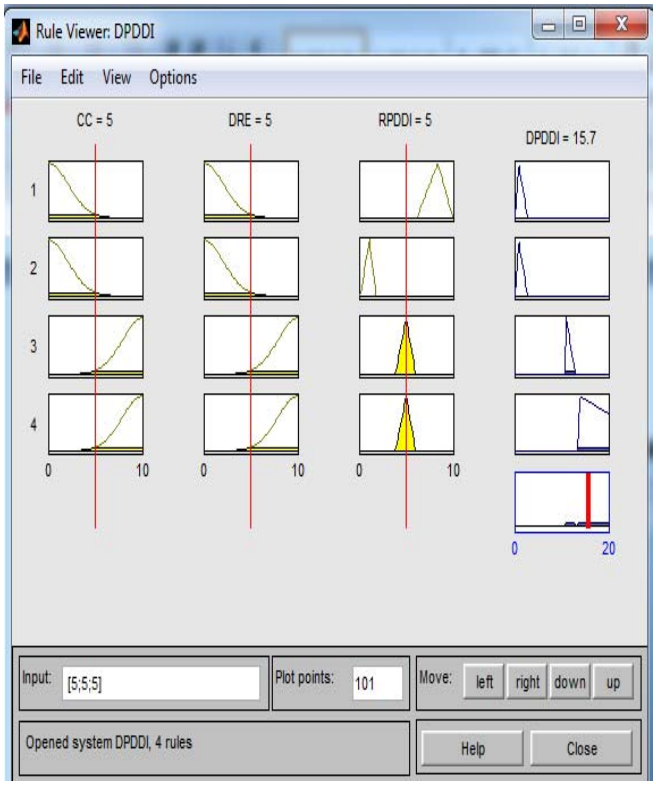


Fig. 3. Fuzzy Rule Viewer for Design phase defect density indicator

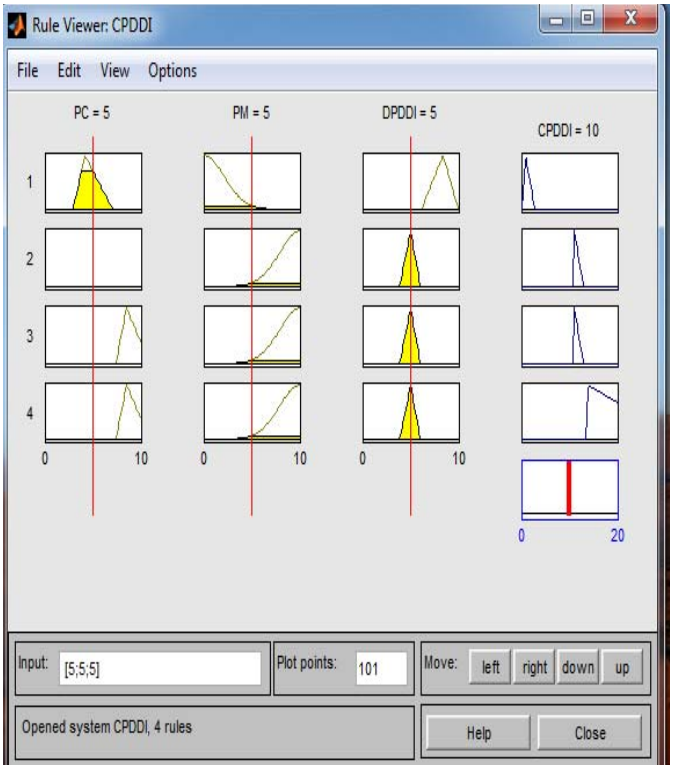


Fig. 4. Fuzzy Logic Rule Viewer for Code phase defect density indicator

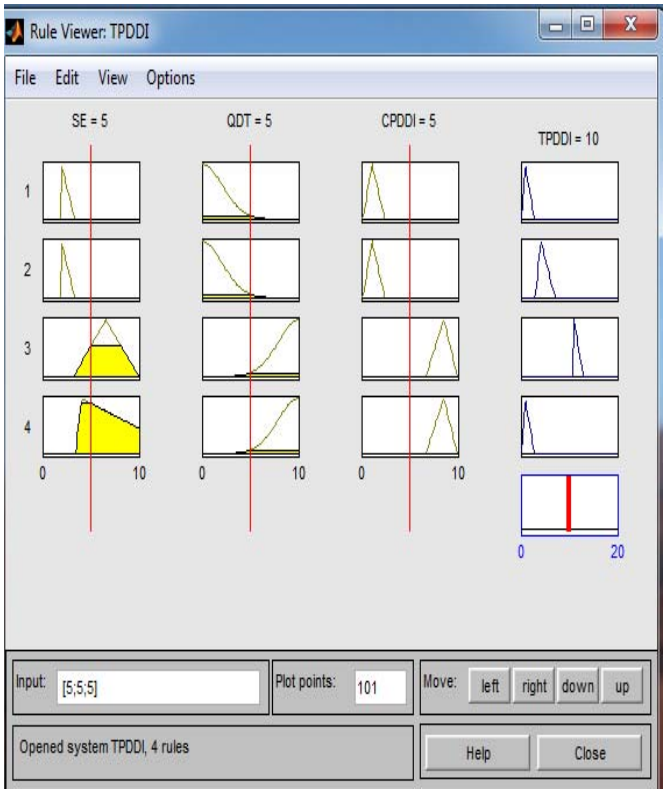


Fig. 5. Fuzzy Rule Viewer for Testing phase defect density indicator

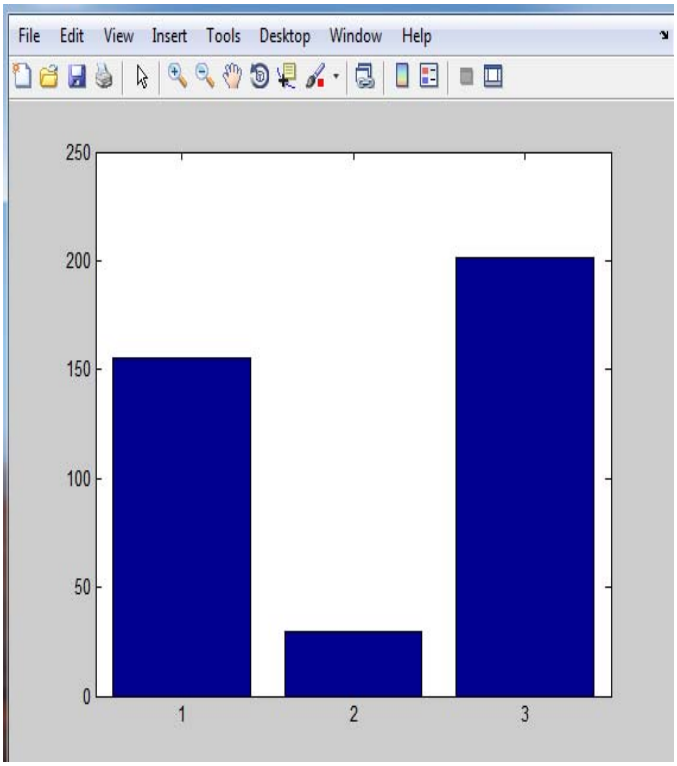


Fig. 6. Fuzzy Modelling for the Implementation of the Existing Work

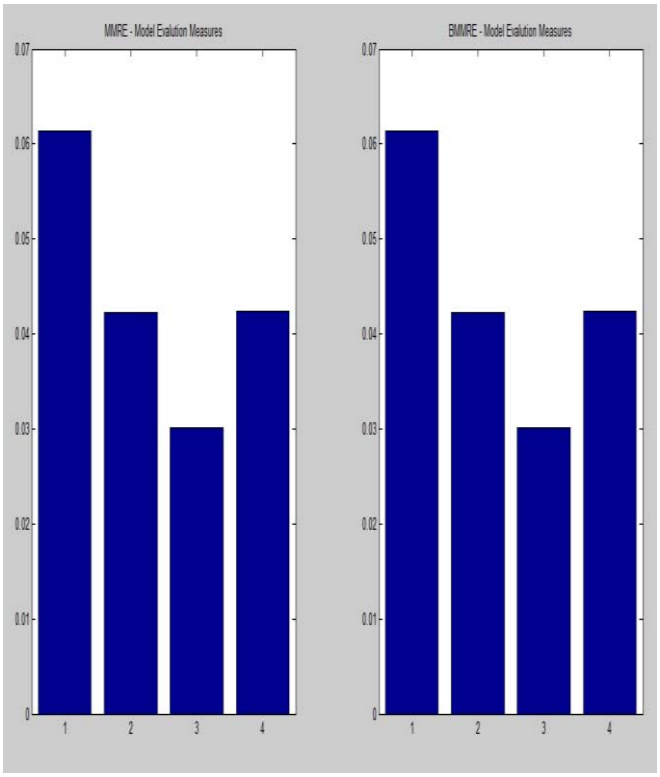


Fig. 7. Evaluation of MMRE and BMMRE Measure

In the proposed implementation for software defects prediction, the integration of MATLAB ANN toolbox is done. Using MATLAB ANN tool, a refined dataset of base work is trained as the dataset on feed forward back propagation network. The data is imported from workspace to MATLAB environment so that ANN tool can use it for creating and training the model of ANN. These values are fetched live from the real world projects used in the base work and analysed using fuzzy modelling. A new feed forward back propagation network is created with the input and target that was fetched from workspace and further predictive model is prepared. The parameter used for training and creating the model is done here so that a predictive analysis of software defects and their threshold can be done. These values and aspects will be used to decide the sample data and other test data sets in determining their acceptability level and software defects anatomy.

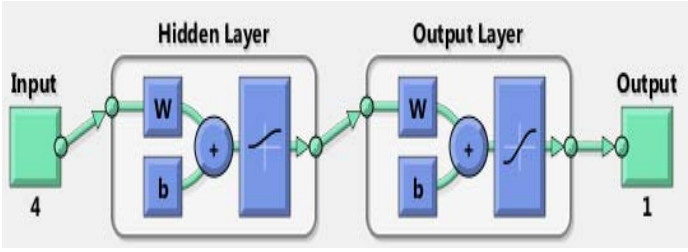


Fig. 8. Structure of ANN

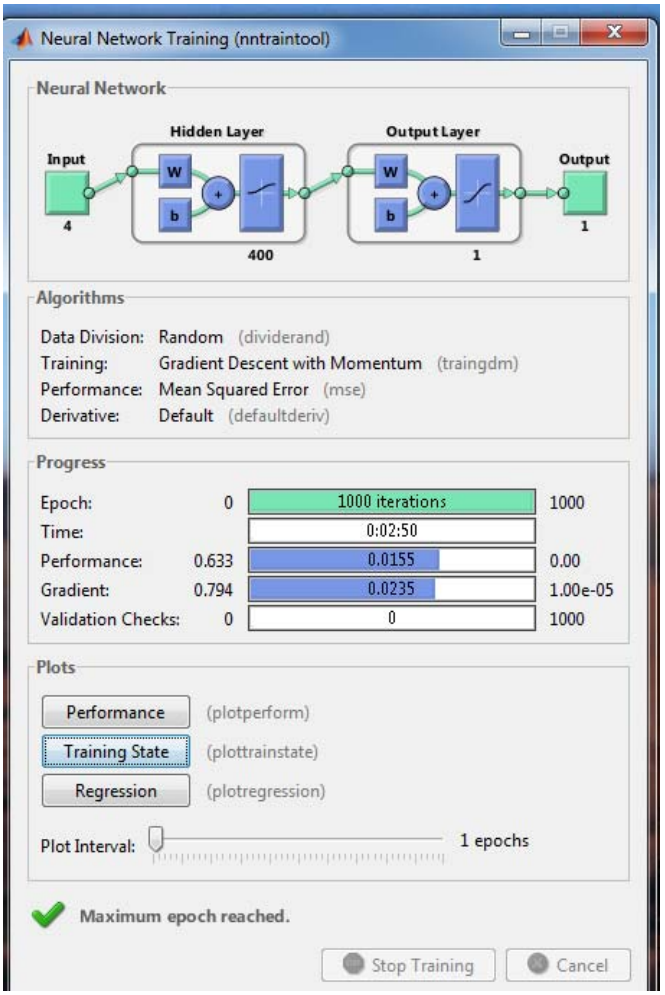


Fig. 9. Network Training Epoch with 1000 iterations

The training of feed forward back propagation model initiate with the specified number of epochs and training parameters.

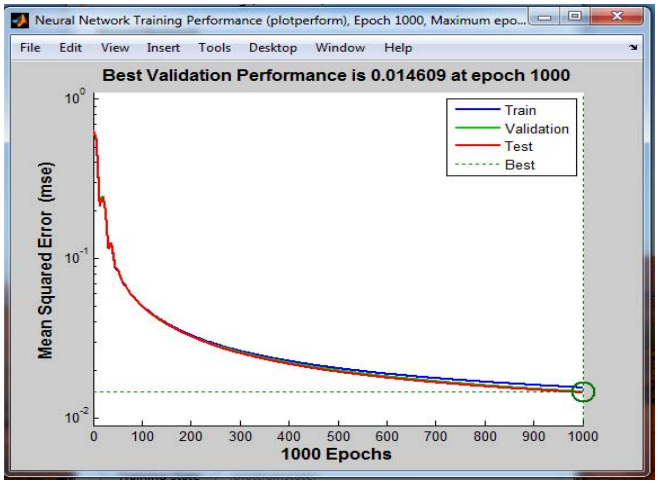


Fig. 10. Network Training Performance

The mean squared error in the above figure obtained after simulation is as per the expectations. The error factor is reducing and towards the zero level determines that the eventual and progressive epochs are giving effective results with very less error rate. At the initial level, the error factor is more which should be towards down line and zero error line.

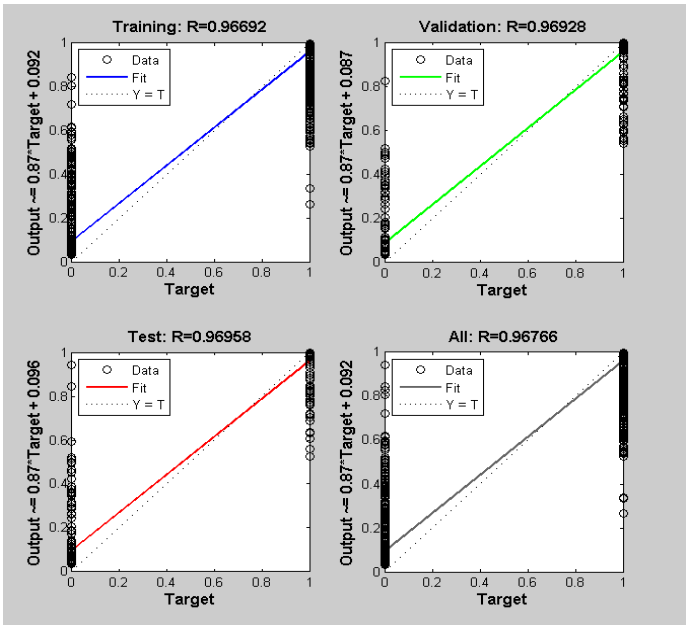


Fig. 11. Error factor

When you run a regression, the tool automatically calculates and plots residuals to help the analyzer to understand and improve the regression model. The line should be very close to linear plot at the middle of graph and shows the minimum error factor.

TABLE 1: Predicted and Actual Defects

Defects in Existing work	Defects in Proposed Work
155	163
30	32
203	207

The proposed approach technique is showing better results as compare to the classical approach.

TABLE 2: Assessment Table

Project size	MMRE		BMMRE	
	Existing Work	Proposed Work	Existing Work	Proposed Work
Project size < 5KLOC (n=5)	0.0613	0.0385	0.0642	0.0375
5KLOC<=Project<50KLOC (n=10)	0.0422	0.0245	0.0412	0.0248
Projects <=50 KLOC (n=5)	0.0301	0.0184	0.0310	0.0123
All Projects (n=20)	0.0423	0.0247	0.0430	0.0244

The error factor in proposed approach (ANN based) is less than the classical approach (fuzzy logic based).

V. CONCLUSIONS

In this research work, a novel, effective and performance aware algorithmic implementation is done with the integration of artificial neural networks and found that the ANN based approach is giving better results than fuzzy logic based approach. The fuzzy logic based implementation is done in the existing work in which a set of rules are mentioned and based on these rules, the software defect prediction metrics can be evaluated. The proposed ANN based approach is effective than the classical fuzzy based approach and giving more accurate and precision based values which are more effective and be used in hybrid approach to a large dataset for better performance aware. In future, following techniques can and efficient results – Decision Tree, Bayesian Net, Random Forest, Honey Bee Algorithm, Swarm Intelligence, Simulated Annealing, Genetic Algorithm

ACKNOWLEDGMENT

The author appreciates the assistance of Dr. Gagandeep an associate professor at Punjabi University Patiala, who helped and guided for this work. Her support, patience, and remarks from time to time greatly helped in completing this work.

REFERENCES

[1] A. Panday Kumar, N. K Goyel. A fuzzy Model for Early Defect prediction and Module ranking. International Journal of Performability Engneering Vol. 8, No. 6, November 2012, pp. 689-698.

[2] C. Kaner and Walter P.Bond. Software Engineering Metrics: What Do They Measure and How Do We Know? 10th International Software Metrics Symposium, Metrics 2004

[3] D. Mundada, Aachal Murade, OmkarVaidya, Swathi J. N. Software fault prediction using artificial neural network and Resilient Back Propagation. International Journal of Computer Science Engineering Vol. 5 No.03 May 2016.

[4] D. kumar Yadav, S. K Chaturvedi and Ravindra B.Misra. Early Software Defects Prediction Using Fuzzy Logic. International Journal of Performability Engineering Vol. 8, No. 4, July 2012, pp. 399-408.

[5] J. Nam. Survey on software Defect Prediction, HKUST PhD Qualifying Examination. 2014.

[6] M. Singh, Dalwinder Singh Salaria. Software Defect Prediction Tol Based On Neural Network, International Journal of Computer Application Vol. 70, No. 22, May 2013, pp. 0975-8887.

[7] M..C. Prasad, Lilly Florence and Arti Arya. A Study on Software Metrics based Software Defect Prediction using Data Mining and Machine Learning Techniques. International Journal of Database Theory and Application. Vol.8, No.3, pp.179-190,2015.

[8] M. Singh Rawat and Sanjay Kumar Dubay. Software Defect Prediction for Quality Improvement. International Journal of Computer Science Vol. 9, No. 2, September 2012.

[9] M. Inayathulla, Silpa C**.** An Approach to Reduce Test Effort by Using Machine Learning Techniques. International Journal of Computer Science and Network Security, VOL.15 No.9, September2015.

[10] N. Fenton, Martin Neil, William Marsh, PeterHearty, ŁukaszRadliński and Paul Krause. On the effectiveness of early life cycle defect prediction with Bayesian Nets. Springer Science and Business Media, June 2008, Editor: Tim Menzies.

[11] Onur Kutlubay, Mehmet Balman, Doğu Gül, Ayşe B. Bener. A Machine Learning Based Model For Software Defect Prediction

[12] P. S.Sandhu, Suman Lata, and Dalveer Kaur Grewal. Neural Network Approach for Software Defect Prediction Based on Quantitative and Qualitative Factors. International Journal of Computer Theory and Engineering Vol. 4, No. 2, April 2012

[13] P. Paramshetti, D. A.Phalke.Survey on Software Defect Prediction Using Machine Learning Techniques. International Journal of Science and Research (IJSR). Vol. 3 Issue 12, December 2014.

[14] Saduf, Mohd Arif Wani, Comparative Study of Back Propagation Learning Algorithms for Neural Networks. International Journal of Advanced Research in Computer Science and Software Engineering. Vol 3, Issue 12, December 2013.

[15] V. Vashish, Manohar Lal, G. S.Sureshchandar. A Framework for Software Defect Prediction Using Neural Networks. Journal of Software Engineering and Applications, 384-394 August 2015

[16] Harikesh Bahadur Yadav, Dilip Kumar Yadav' .A fuzzy logic based approach for phase-wise software defects prediction using software metrics. Information and Software Technology .Vol. 63, July 2015, Pages 44–57.