APPLICATION OF MACHINE LEARNING IN PREDICTING SOFTWARE QUALITY USING NEURO-FUZZY SYSTEM

Ashish Raturi, Preeti Malik

Scientist-B
National Institute of Electronics and Information Technology, Delhi, India

Abstract: Predicting quality of software is one of the major concerns in software engineering that needs to be estimated. It can be predicted effectively with the help of combination of neural networks plus fuzzy logic which is collectively referred as Neuro-Fuzzy technique. *It is one of the research areas of machine learning.* The Neuro-fuzzy model proposed in paper is used to estimate number of defects/faults in system by taking object oriented (OO) design metrics into consideration design metrics are used as independent variables while numbers of faults is used as dependent variable.
The paper conducts empirical study of model by taking public dataset KCI from NASA project database and the model is being validated by using MATLAB simulation.

Keywords: - Software quality, OO Metrics, FuzzyInference System (FIS) and Neuro-Fuzzy Technique

I. Introduction

The changing business rules of software industry have led to increase in complexity of software that causes uncertainty among users problem domain. Various parametric techniques have been failed to handle such uncertainties. They are been handled by soft computing techniques like neural networks (NN), neuro-fuzzy techniques (ANFIS), machine learning algorithm and estimations by analogy. neural networks has its application in areas like finance, geophysics, medicine, geology etc. whenever there is task of prediction, control or classification, the solution is provided by combination of neural networks and fuzzy logic system. It is not feasible to design full fault prone system due to changing business requirements and complexity of software. Various studies have been conducted and still going in order to estimate or product number of faults in system by incorporating several novel techniques for measurement of object oriented (OO)design characteristics like encapsulation, inheritance, information hiding and message passing. The objective of study defines proposed neuro-fuzzy models that predicts error rate by validating it on public KCI dataset under NASA object database. The results are shown with help of simulation tool called MATLAB. The remainder of paper is organized as follows: Section 2 presents back round of conducted studies by various researchers. Section 3 gives brief description of fuzzy logic. Section 4 conducts empirical study by processing and analyzing dataset using MATLAB. It is followed by conclusion and references.

II. Literature Review

Khoshgaftar et.al[1] devised model for predicting software quality by using neural networks (NN) they compared neural network with discriminate model and found NN model has better accuracy. But, it did not make use fuzzy rules to discover explicit knowledge related to model. Kamiyaet.al[2] proposed a method to estimate number of faults in given closes by using complexity metrics as independent variables. Emamet.al [3] devised model to predict faulty classes in java application. Fioravanti and Nesi [4] worked on. 200 different metrics for identifying fault proneness among classes. Aggarwal et.al [5] propped furry model to predict software quality. It is based on furry logic approach that identifies error prone components. Singh et.al [6] proposed Gene Expression Programming (GEP) algorithm by using soft computing technique. It is used to product fault proneness software by introducing new metric fault factor (FF). Hu and Zhong [7] prepared model to predict risks associated with software module using NN.

III. Fuzzy Inference System (FIS)

Estimating number of faults for different iterations cannot be done by traditional methods that rely only on algorithmic data without deriving any knowledge. To perform this task, two machine learning techniques viz. neural network and fuzzy logic system can be used. Both are separate techniques with their different roles but the model proposed in study comprises of neuro-fuzzy hybrid system that combines advantages of fuzzy system (for deriving exploit knowledge) and neural network approach (for deriving implicit knowledge).

Fuzzy logic system or fuzzy inference system (FIS) is a technique that works on human decisions by deriving inferences from vague and ambiguous data. It applies knowledge gained through experience and fuzzy mathematics to solve prediction issues. It is known that FIS does not require any prior knowledge and experience related to given problem FIS is rule based system that performs non-linear mapping between input and output. The equation dealing with fuzzy system is given by:-

$f(x) = \sum(i=1 \text{ to } m) y^i(\pi(i=1 \text{ to } n) F(x_i) / \sum(i=1 \text{ to } m)\ \pi(i=1 \text{ to } n) F(x_i)$**[8]**

The parameters used in equation are as:-

y→ output

i→ 1 to m means rules from 1 to m

$\times_i$= input set (x_1, x_2.......................x_m)

π = membership function

F(x)= individual function values associated with rules

IV. Design of Empirical Study

The study is conducted in order to estimate number of faults present in given classes. Traditional metrics are not adequate for estimating OO faults. So, object oriented CK metrics are used that have objected oriented features like inheritance, coupling, memory allocation etc. The word CK stands for Chidamber and Kemerer **[**9**].**

4.1. Source of data

The study involves usage of public dataset KC1 from NASA IV and V Facility Metrics Data Program Repository (MDP) **[**10**].** The dataset consists of 43 KSLOC of C++ code with 145 classes and 2107 instances.

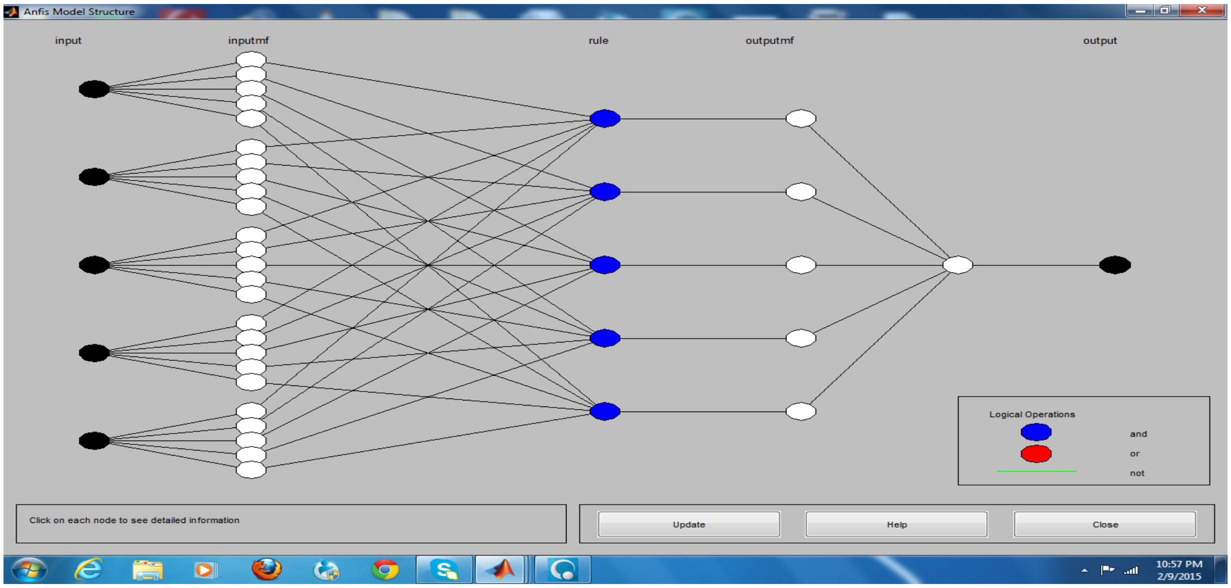
Table 1: Descriptive statistics of 145 studied C++ classes of KC1 **[**11**]**

Metrics	Min	Max	Mean	Mode	S.D.	Median
WMC	0	110	17.42	8	17.45	12
RFC	0	222	34.38	7	36.20	28
CBO	0	24	8.32	0	6.34	8
DIT	1	7	2	1	1.26	2
NOC	0	5	0.20	0	0.70	0
LCOM	0	100	68.71	1	36.89	84

4.2. Evaluation Criteria

The training data shown above is used to identify parameters for OO model that deals with lagged variables with fuzzy sets as input and estimates quality of software. The results are shown with the help of MATLAB using neuro-fuzzy editor.

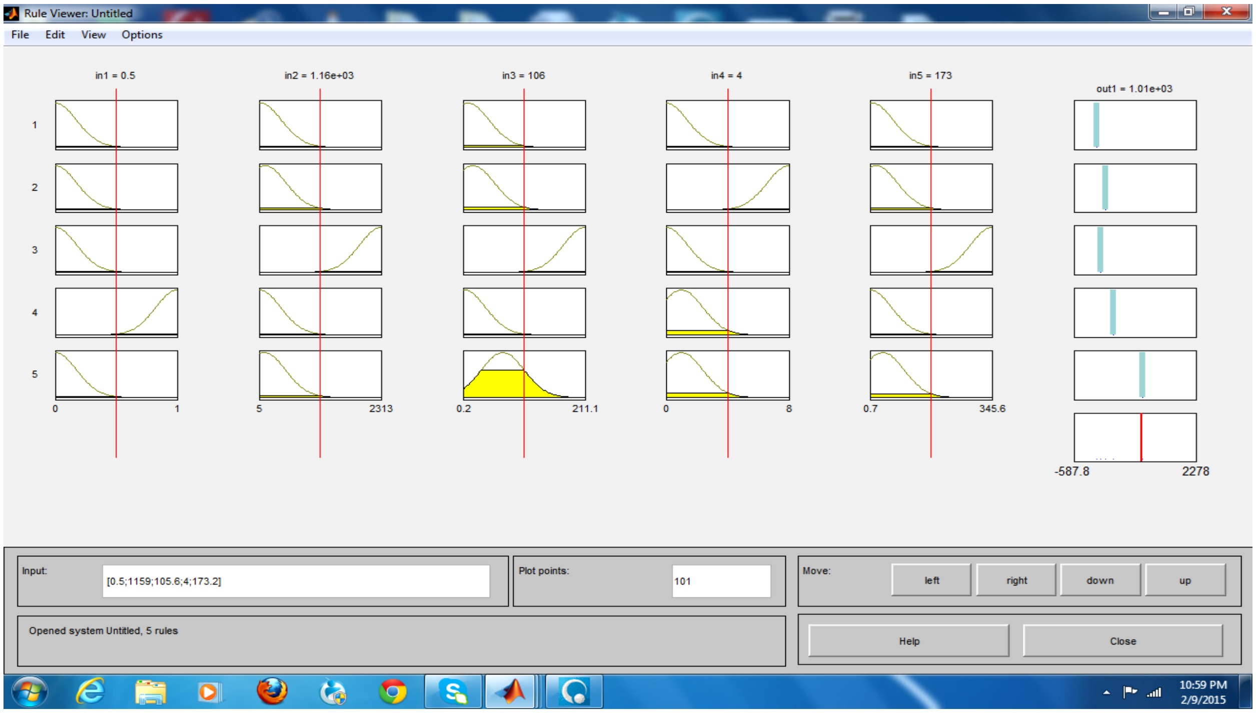
4.3. Results and Discussions



Click on each node to see detailed information

Fig 1: ANFIS structure

It depicts five rules that are produced on basis of input functions. The rules are represented by “AND” function.



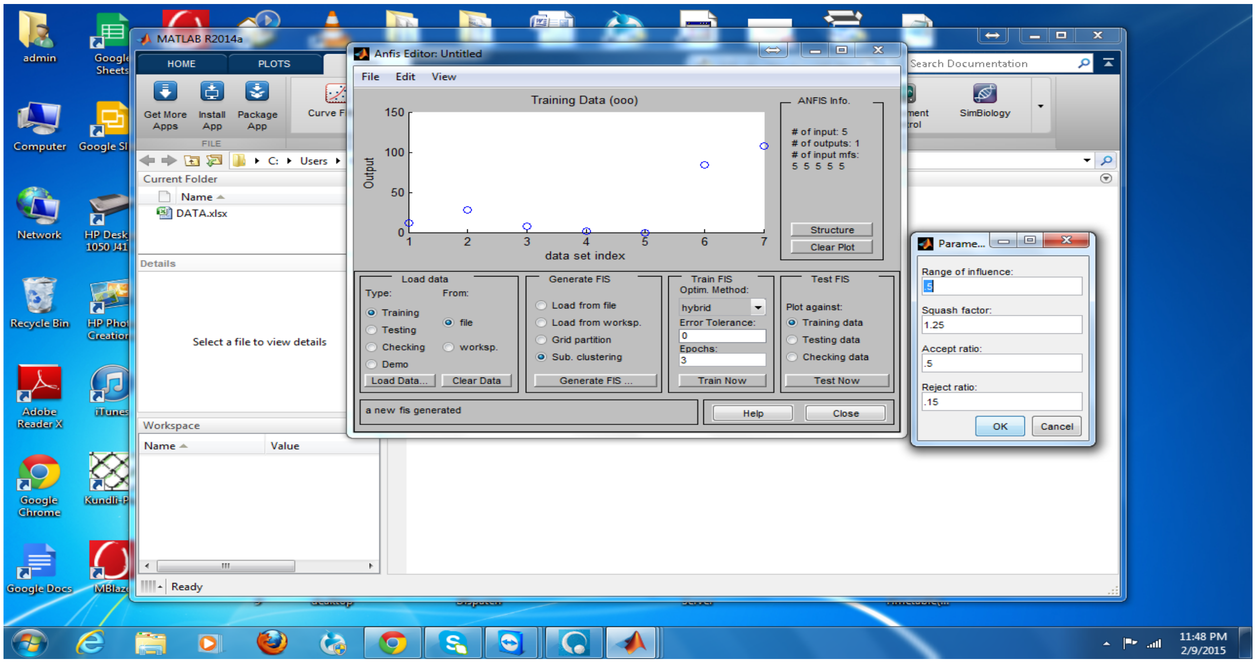
Input: [0.5;1159;105.6;4;173.2]

Plot points: 101

Opened system Untitled, 5 rules

Fig 2: FIS set of rules

Above figure shows five rules with their corresponding values that are clubbed in form of ANFIS structure in figure 3. The rules are represented by their individual functions as in1=0.5, in2=1.16e+0.3, in3=106, in4=4 and in5=173 with output as out1=1.01e+0.3



# of input: 5
# of outputs: 1
# of input mfs:
5 5 5 5 5

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ● file ○ worksp.

● Sub. clustering

Optim. Method: hybrid

Error Tolerance: 0

Epochs: 3

Plot against: ● Training data ○ Testing data ○ Checking data

a new fis generated

Range of influence: .5

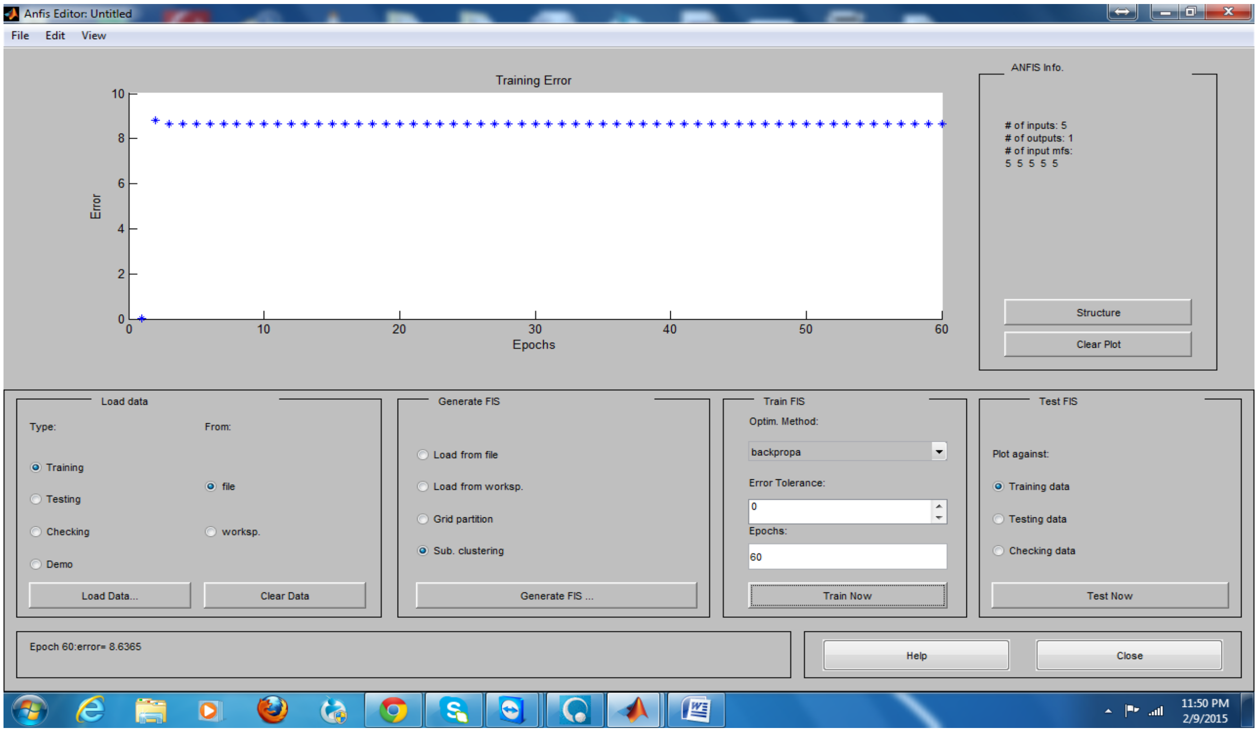
Squash factor: 1.25

Accept ratio: .5

Reject ratio: .15

Fig 3: Training dataset

The actual output is being represented by plotting training data graph. It consists information about ANFIS structure holding 5 inputs and 1 output with 5 input functions. It uses hybrid method with error tolerance factor set to zero.



ANFIS Info.

# of inputs: 5
# of outputs: 1
# of input mfs:
5 5 5 5 5

Load data

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ● file ○ worksp.

Generate FIS

○ Load from file
○ Load from worksp.
○ Grid partition
● Sub. clustering

Train FIS

Optim. Method: backpropa

Error Tolerance: 0

Epochs: 60

Test FIS

Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 60:error= 8.6365

Fig 4: Plot of training data vs. epochs

V. Conclusion

The paper proposed neuro-fuzzy model that is combination of two machine learning techniques-fuzzy logic and neural networks. Both are combined in order to derive explicit and implicit knowledge. The objective of study defines proposed neuro-fuzzy models that predicts error rate by validating it on public KCI dataset under NASA object database. The study is conducted to predict error rate in proposed model and compared it with statistical techniques. It is done by simulation tool called MATLAB. From the results presented above, it is said that Object oriented (OO) metrics appear to be useful to predict number of faults. The training dataset is used and graphs are plotted between actual and expected output.

References

[1]. T.M. Khoshgoftaar, E.B. Allen, J.P. Hudepohl, and S.J.Aud, “Application of neural networks to software quality modeling of a very large telecommunications system”, IEEE Transactions on Neural Network, vol. 8, pp. 902-909, 1997

[2]. T. Kamiya, Kusumoto, S., K. Inoue,”Prediction of faultproneness at early phase in object-oriented development”, *Proceedings of the Second IEEE International Symposium onObject-Oriented Real-Time Distributed Computing*, pp., 253- 258, 1999

[3]. E.L. Emam, W. Melo and C.M. Javam ―The Prediction of Faulty Classes Using Object-Oriented Design Metrics‖, Journal of Systems and Software, Elsevier Science, pp. 63-75, 2001

[4]. F. Fioravanti and P. Nesi ―A study on fault-proneness detection of object-oriented systems‖, Fifth European Conference on Software Maintenance and Reengineering, pp. 121 –130, 2001

[5]. Aggarwal, K.K., Singh, Y., Chandra, P., and Puri, M., Measurement of Software Maintainability Using a Fuzzy Model, Journal of Computer Sciences, vol. 1, no.4, pp. 538-542, 2005

[6]. Singh, Y., Kaur, A., and Malhotra, R., Prediction of software quality model using Gene Expression Programming, Proceedings of the 10th international Conference on Product-Focused Software Process Improvement (PROFES 2009), Lecture Notes in Business Information Processing, vol. 32, springer-verlag, pp. 43-58, 2009

[7]. Y. Zhao, C. Zhong, Z. Li, T. Yan, ―Object-Oriented Software Fault-Proneness Prediction Using Support Vector Machine‖ (in Chinese), Computer Engineering & Science, Vol.30, No.11, pp.115-117, 2008

[8]. Yetim, Yavuz, Martonosi, Margaret, Malik, Sharad. Extracting useful computation from error-prone processors for streaming applications. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013

[9]. Chidamber, Shyam R., and Chris F. Kemerer. "A metrics suite for object oriented design." Software Engineering, IEEE Transactions on 20.6 (1994): 476-49

[10]. Sayyad Shirabad, J. and Menzies, T.J. (2005) The PROMISE Repository of Software Engineering Databases., School of Information Technology and Engineering, University of Ottawa, Canada. Available: http://promise.site.uottawa.ca/SERepository

[11]. http://promise.site.uottawa.ca/SERepository/datasets/kc1.arff

[12]. Mie MieThetThwin and Tong Seng Quah (2002). Application of neural network for predicting software development faults using object oriented design metrics. Proceedings of the 9th international conference on neural information processing (ICONIP,02), Vol.5

[13]. Thomas J. Ostrand et.al. Predicting bugs in large industrial software systems. Software engineering lecture notes in computer science, volume 7171, 2013, pp 71-93

[14]. BerkhanDeniz, SemihBilgen. An empirical study of software reuse and quality in an industrial setting. Lecture notes in computer science, Computer science and applications (ICCSA) 2014, vol. 8583, pp 508-523.

[15]. Panagiotidis, A., Reina, G. Ertl, T. Strategies for Fault-Tolerant Distributed Visualization. Pacific Visualization Symposium (PacificVis), 2014 IEEE, pp 286-290