Software Effort Estimation for Size Proxy Metric Framework Modelling using Software Estimation Models and Neuro Fuzzy Logic Approach

Shama Kousar Jabeen.A, B.Arthi

Abstract— As software grew in size and requirements it also successively grew in complexity and cost. Evaluating size estimates accurately at an initial stage in the software conglomeration is of high priority. Conventional techniques have the problems of uncertainty and precision during the evaluation of size estimates. Software engineering cost models and estimation techniques are used for a number of purposes. In our work we have compared the results using three function point based effort estimation models. We have also compared MMRE, MMER, MRE, MER values by training the dataset using neuro-fuzzy logic based machine learning aapproach which overcomes the problems present in the traditional methods. In this paper effort estimates has been obtained by modeling and training the size metric framework. The dataset trained in our work is for 100 projects .

Keywords—size metric, fuzzy logic software effort ,software engineering, cost estimation models, MMRE ,MER,MRE, MMER.

I. INTRODUCTION

Delivering the software on time and within budget is a critical concern for many organizations Cost estimations refers to the prediction in terms of time, staff, and effort. In many papers cost estimation is referred to as the effort prediction and hence it is used interchangeably. Effort prediction is usually made at an early stage of software development. Difficulty prevails in estimation because the estimates are very often uncertain and little knowledge is known. The size proxy metric framework considers the mean and variance of effort which is not considered in traditional metrics. In effort prediction laziness and ignorance must be taken in to account as the software designer does not consider all the factors during the prediction process. We consider uncertainty in our work which is considerable across specific project size metrics. The work has been implemented with a framework where the size metric is considered for better effort prediction. The entire project in this paper has been implemented in two major parts. The first part consists of extraction of function points from requirements document. In the second part the extracted function points are used to calculate effort estimates by training the dataset. This training of the dataset has been done using fuzzy logic approach.

This paper is organized as follows. Section 2 discusses about the related works. Architecture implementation and the concepts carried out in our work has been discussed in Section 3.Section 4 shows the experimental results and the implementation of the system. Section 5 draws conclusions and the future work.

Manuscript received May 2014

Shama Kousar Jabeen.A, Department Of Information Technology, Easwari Engineering College, Ramapuram, Chennai, India.

B.Arthi, Department Of Information Technology, Easwari Engineering College, Ramapuram, Chennai, India.

II. RELATED WORKS

Boehm's[1] describes that ,software engineering practices in the industry determines the cost and the quality of the software product .Thus a large and increasingly costly item also makes a large and increasing impact on human welfare.

The work done by Moataz A.Ahamed, Irfan Ahmad and Jarallah S.Alghamdi[9] explains the size proxy framework which considers normal distribution in combination with the regression model .They have considered LOC as inputs for early software effort prediction. There are basically two areas of research dealt in effort prediction: (1) developing prediction techniques and models, (2) developing size metrics that could be used as a proxy for effort estimation.

B. Boehm, C. Abts, S. Chulani [2] have illustrated in their work about the first the area of research .In this paper, different methodologies of effort prediction like expert judgment, analogy based prediction, algorithmic models, non algorithmic approaches have been detailed. Expert judgment is a time consuming approach. Prediction using analogy requires a predetermined effort estimate which is then used to compare similar or analogous projects.

Algorithmic models are many which involves COCOMO, SLIM, SEER-SEM models. Non algorithmic modeling methods are based on machine learning and soft computing techniques. Some of them are Bayesian belief networks, fuzzy logic approach, artificial neural networks, and evolutionary computation. There are also researches done using soft computing approaches in combinations, such as neuro-fuzzy, neuro-genetic approaches.

Moataz A.Ahamed and Zeeshan Muzaffar[10] have suggested that traditional approaches for software projects effort prediction such as the use of the mathematical formulae are derived from the historical data ,or the use of expert judgment are plagued with issues pertaining to effectiveness and robustness in their results. Thus type-II fuzzy logic systems must be allowed to handle imprecision and uncertainty. They have considered COCOMO model in their work.

The second area of research in effort prediction is to develop early and better size metrics for a good software effort prediction. Some of them which involves usecases, class diagrams, source code. The most frequently and commonly used size metrics in the early phases of software development lifecycle are LOC (lines of code) and FP (Function points), project features and use cases.

The function point metric was proposed by Albrecht as a method for measuring software size and productivity. Function point metric sizes software from the end user perspective by measuring the functionality delivered to the end user .

A function point is defined as one end user business function. It employs functional and logical entities such as inputs, outputs, files and inquiries that are believed to relate more closely to the functions performed by the software as compared to other measures such as lines of code.
The counting of function points is based on IFPUG[3] as suggested in their standards. There are many other derivatives of function point metric proposed in the literature trying to address and overcome issues related to using function point metric measures size. Mark II function point metric is one of the widely used in the industry.
Justin Wong,Danny Ho,Luiz Fernando,Capretz[4] have suggested that the neuro fuzzy function point backfiring (NFFPB) makes use of neural network which is used for tuning the fuzzy logic membership functions where backfiring approach was used .In this method function points are converted in to SLOC estimates and programming languages where grouped based on the fuzzy levels.
K.K.Shula[6] has discussed in his work on the substantial improvement in the prediction accuracy by the neuro genetic approach as compared to both a regression tree based conventional approach .They have made use of COCOMO data set comprising of 63 projects and kemerer data set comprising 15 projects which were merged randomly in to a single database of 18 projects.

III. PROPOSED METHODOLOGY

The work in this paper has been done for size estimation using ANFIS toolbox in order to predict the overall effort of a software system. As a project's effort estimate is obtained it is used to evaluate the size estimate of the software system. Based upon the size, the customer approves or rejects the proposal of implementation.
The framework which has been cited in the work of M.A.Ahamed, Irfan Ahmad, and Jarallah S.Alghamdi[9] is used as a proxy to train the size estimates. We have considered this paper as a major source for our work. Since there work has the problem of uncertainty ,we overcome this issue with the help of soft computing approach like fuzzy logic in combination with neuro fuzzy system which is highly suitable to train the size proxy.
Justin Wang, Danny ho,Luiz Fernando Capretz[4] suggested that Neuro fuzzy is a technique that integrates neural network and fuzzy logic. K.Srinivasan and D.Fisher[5] have stated the reason for integration, that the technique takes advantage sf the neural networks learning capability and fuzzy logic's human like reasoning. Moreover the training of the dataset will also lead to better and accurate estimates. Henceforth, we have carried out our work using ANFIS toolbox in MATLAB and function points size estimates obtained from the different software models. This results in precision and accuracy of effort value by the development of suitable framework to illuminate , size proxy metric for effort prediction.
Roger.S.Pressman[11] has suggested the overall structure of software models as follows:

$$E=A+B*(ev)^c \qquad (1)$$

Where A,B,C are empirical constants. E is effort in person months and estimation variable (either LOC or FP).In addition to the above equation he also suggests effort equation(2), equation(3), equation(4) to estimate effort from function points using various FP oriented models as given below:

$$E=-91.4+0.355FP \qquad (2)$$

$$E=-37+0.96FP \qquad (3)$$

$$E=-12.88+0.405FP \qquad (4)$$

Here equation(2) is Albrecht and Gaffney model. Equation(3) is kemerer model. Equation(4) is small project regression model. A quick examination of these models indicate that each will yield a different result for the same values of LOC or FP .The implications is clear that the estimation models must be calibrated for local needs of our estimation. In our work we have implemented all the three models for our effort estimation in terms of size which has been feeded as input to ANFIS editor for training the dataset.
L.Putnam and W.Myers[7] have suggested four methods to overcome sizing problem as follows:

1) Fuzzy Logic Sizing approach: In this approach the project manager must identify the application nature and type to consequently establish it's magnitude within the original range.
2) Function point Sizing approach: The requirements engineer develops estimates of the information domain features.
(3) Component Sizing approach: The software application is composed of various types of components that are unique to a particular application area.
4) Change sizing approach: This method is used when a project circumscribes the usage of software that must be modified as a subdivision of a software project.

The system architecture implemented in our work is as shown below in figure1 .It is split up in to two larger modules. First module, deals with generation of size estimates from the requirements document. The second module deals with training of the dataset using ANFIS editor in our fuzzy logic toolbox in MATLAB R2009a.The first module deals with the calculation of function points using the three FP oriented estimation models. Once the FP is estimated it is fed in to the ANFIS as a data file. The first step in the ANFIS training module deals with the loading of the training data. Then it is followed by generation of FIS where we have made use of a sugeno fuzzy logic systems to generate crisp outputs. Then the generated FIS has been trained which is finally compared against original training data in order to assess the testing error. Thus in the FIS effort is obtained as a triangular membership function with 3 linguistic variables and 27 rules.

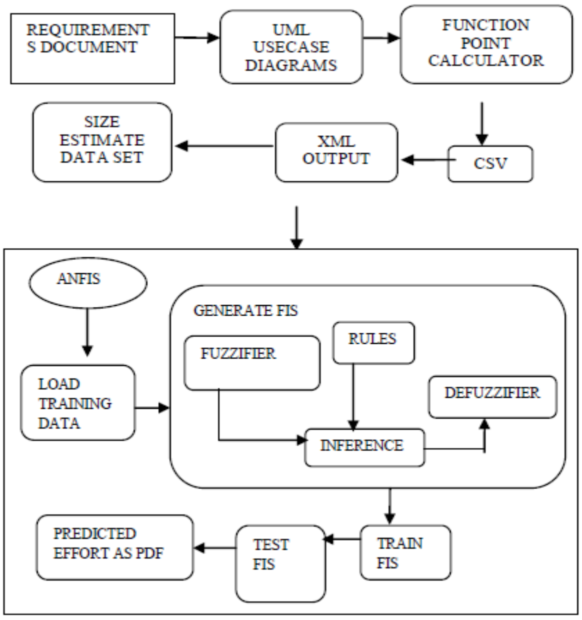

Figure1:System Architecture

In function point estimation process, decomposition work is canonical. The subsequent estimation process does not account only for functionality, but rather it focuses on the domain characteristics and complexity issues as well. The resultant estimates can then be used to obtain a FP value that can be correlated to the past data and used to achieve a general estimate. The work has involved calculation of function points by extracting use cases from the requirements document. This work has been implemented using Visual Use case design tool [11] where the usecases are imported from the requirements document. The usecase diagrams are rated as simple, average or complex . This obtains the unadjusted function point count.

Stephen H. Kan[12] refers to the calculation of value adjustment factors(VAF) which forms the second part of function point computation. This is done using 14 general system characteristics (GSC'S). These are rated in a scale of 0 to 5 in order to assess their impact .The 14 GSC's are as follows:

1) Data Communication functions.
2) Distributed functions in system.
3) Performance of the system.
4) Heavily used configuration of the system.
5) Transaction Rate of the system.
6) Online data entry of the system.
7) End user efficiency of the system.
8) Online Update in system.
9) Complex Processing of the system.
10) Reusability of the system.
11) Installation Ease of the system.
12) Operational Ease of the system.
13) Multiple sites usage
14) Facilitation of Change of the system..

The unadjusted function point depends upon the complexity judgment of the software application in terms of five components which are:

1) External Input.
2) External Output.
3) Logical internal file.
4) External Interface file.
5) External inquiry

Stephen H.Kan[12] has cited equation(5) to calculate VAF as shown below. This equation has been used in this paper to rate the complexities of the usecase diagrams used in our study dataset.

$$VAF=0.65+0.01\sum_{i=1}^{i=14} c_i \tag{5}$$

Stephen H.Kan[12] has referred to equation(6) to calculate function points as shown below. This equation has been implemented in our work to estimate function points for the author's study dataset.

$$FP=FC*VAF \tag{6}$$

Equation(6) has been used in this work which is a simple derivation used in the calculation of function points. A project manager or a software analyst must understand the full documented methods such as International Function Point User's Group (IFPUG)[3] standard for a complete and accurate implementation. Thus the size estimated, dataset is then applied to Sugeno type systems.

M.Wasif Nasar, Yong-Ji Wang and Manzoor Elahi[8] in their work have expressed about fuzzy logic as a mathematical tool for dealing with uncertainty and imprecision. It is a theory of unsharp boundaries and is used to solve problems that are too complex to be understood qualitatively .The concept of fuzzy sets be viewed as a generalization of the concept of a classical crisp set.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In our work, we have proposed function point oriented effort estimation model rules which contain linguistic variables related to the project. The rule base for fuzzy inference system (FIS) integrated in the ANFIS toolbox makes use of OR ,AND logical operation with unadjusted function point, value adjustment factor and function points as input variables to form a large number of rules in our work. We have implemented our work with 3 inputs and 1 output resulting in a combination of 27 rules. The rules are implemented individually for each of the FP oriented estimation models. After estimating the effort values other parameters such as MMRE,MMER,MRE and MER are found out and they are being compared in order to conclude the best model suiting our local dataset The fuzzy logic rules for OR operation are as follows:

Pseudo code:

Rule 1: If (UFC is low) or (VAF is low) or (FP is low) then (EFFORT is low) (1).
Rule 2: If (UFC is low) or (VAF is low) or (FP is medium) then (EFFORT is medium) (1).
Rule 3: If (UFC is low) or (VAF is low) or (FP is high) then (EFFORT is high) (1).
Rule 4: If (UFC is low) or (VAF is high) or (FP is low) then (EFFORT is high) (1).
Rule 5: If (UFC is low) or (VAF is high) or (FP is medium) then (EFFORT is high) (1).
Rule 6: If (UFC is low) or (VAF is high) or (FP is high) then (EFFORT is high) (1).
Rule 7: If (UFC is low) or (VAF is medium) or (FP is low) then (EFFORT is medium) (1).

Rule 8: If (UFC is low) or (VAF is medium) or (FP is medium) then (EFFORT is medium) (1).
Rule 10: If (UFC is medium) or (VAF is low) or (FP is low) then (EFFORT is medium) (1).
Rule 11: If (UFC is medium) or (VAF is low) or (FP is medium) then (EFFORT is medium) (1).
Rule 12: If (UFC is medium) or (VAF is low) or (FP is high) then (EFFORT is high) (1).
Rule 13: If (UFC is medium) or (VAF is medium) or (FP is low) then (EFFORT is medium) (1).
Rule 14: If (UFC is medium) or (VAF is medium) or (FP is medium) then (EFFORT is medium) (1).
Rule 15: If (UFC is medium) or (VAF is medium) or (FP is high) then (EFFORT is high) (1).
Rule 16: If (UFC is medium) or (VAF is high) or (FP is low) then (EFFORT is high) (1).
Rule 17: If (UFC is medium) or (VAF is high) or (FP is medium) then (EFFORT is high) (1).
Rule 18: If (UFC is medium) or (VAF is high) or (FP is high) then (EFFORT is high) (1).
Rule 19: If (UFC is high) or (VAF is low) or (FP is low) then (EFFORT is high) (1).
Rule 20: If (UFC is high) or (VAF is low) or (FP is medium) then (EFFORT is high) (1).
Rule 21: If (UFC is high) or (VAF is low) or (FP is high) then (EFFORT is high) (1).
Rule 22: If (UFC is high) or (VAF is medium) or (FP is low) then (EFFORT is high) (1).
Rule 23: If (UFC is high) or (VAF is medium) or (FP is medium) then (EFFORT is high) (1).
Rule 24: If (UFC is high) or (VAF is medium) or (FP is high) then (EFFORT is high) (1).
Rule 25: If (UFC is high) or (VAF is high) or (FP is low) then (EFFORT is high) (1).
Rule 26: If (UFC is high) or (VAF is high) or (FP is medium) then (EFFORT is high) (1).
Rule 27: If (UFC is high) or (VAF is high) or (FP is high) then (EFFORT is high) (1).
Similarly the fuzzy logic rules for AND operation are as follows:

Pseudo code:

Rule 1: If (UFC is low) and (VAF is low) and (FP is low) then (EFFORT is low) (1).
Rule 2: If (UFC is low) and (VAF is low) and (FP is medium) then (EFFORT is low) (1).
Rule 3: If (UFC is low) and (VAF is low) and (FP is high) then (EFFORT is low) (1).
Rule 4: If (UFC is low) and (VAF is high) and (FP is low) then (EFFORT is low) (1).
Rule 5: If (UFC is low) and (VAF is high) and (FP is medium) then (EFFORT is low) (1).
Rule 6: If (UFC is low) and (VAF is high) and (FP is high) then (EFFORT is low) (1).
Rule 7: If (UFC is low) and (VAF is medium) and (FP is low) then (EFFORT is low) (1).
Rule 8: If (UFC is low) and (VAF is medium) and (FP is medium) then (EFFORT is low) (1).
Rule 9: If (UFC is low) and (VAF is medium) and (FP is high) then (EFFORT is low) (1).
Rule 10: If (UFC is medium) and (VAF is low) and (FP is low) then (EFFORT is low) (1).
Rule 11: If (UFC is medium) and (VAF is low) and (FP is medium) then (EFFORT is low) (1).
Rule 12: If (UFC is medium) and (VAF is low) and (FP is high) then (EFFORT is low) (1).
Rule 13: If (UFC is medium) and (VAF is medium) and (FP is low) then (EFFORT is low) (1).
Rule 14: If (UFC is medium) and (VAF is medium) and (FP is medium) then (EFFORT is medium) (1).
Rule 15: If (UFC is medium) and (VAF is medium) and (FP is high) then (EFFORT is medium) (1).
Rule 16: If (UFC is medium) and (VAF is high) and (FP is low) then (EFFORT is low) (1).
Rule 17: If (UFC is medium) and (VAF is high) and (FP is medium) then (EFFORT is medium) (1).
Rule 18: If (UFC is medium) and (VAF is high) and (FP is high) then (EFFORT is medium) (1).
Rule 19: If (UFC is high) and (VAF is low) and (FP is low) then (EFFORT is low) (1).
Rule 20: If (UFC is high) and (VAF is low) and (FP is medium) then (EFFORT is low) (1).
Rule 21: If (UFC is high) and (VAF is low) and (FP is high) then (EFFORT is low) (1).
Rule 22: If (UFC is high) and (VAF is medium) and (FP is low) then (EFFORT is low) (1).
Rule 23: If (UFC is high) and (VAF is medium) and (FP is medium) then (EFFORT is medium) (1).
Rule 24: If (UFC is high) and (VAF is medium) and (FP is high) then (EFFORT is medium) (1).
Rule 25: If (UFC is high) and (VAF is high) and (FP is low) then (EFFORT is low) (1).
Rule 26: If (UFC is high) and (VAF is high) and (FP is medium) then (EFFORT is medium) (1).
Rule 27: If (UFC is high) and (VAF is high) and (FP is high) then (EFFORT is high) (1).

Shama Kousar Jabeen.A and Mrs. B.Arthi[15] have shown in there work for effort estimation using function points with training the dataset using fuzzy logic approach. Shama Kousar Jabeen.A and Mrs.B.Arthi[14] have shown in there another work for size estimation with neuro fuzzy logic approach for Albrecht dataset alone..This paper is an extension work of Shama Kousar Jabeen.A and Mrs.B.Arthi[14] where the other two FP oriented models have been implemented in ANFIS and there corresponding MMRE,MER,MRE and MMER values have been compared here along with the Albrecht oriented FP Model.

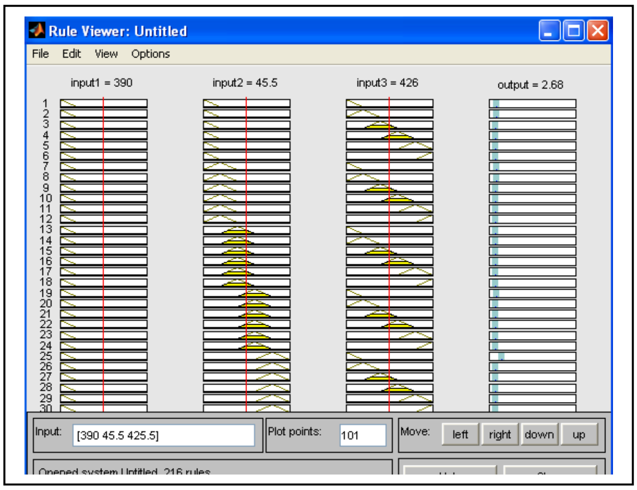

Figure2:Rule Viewer For Albrecht Model, AND Operation In Anfis Editor

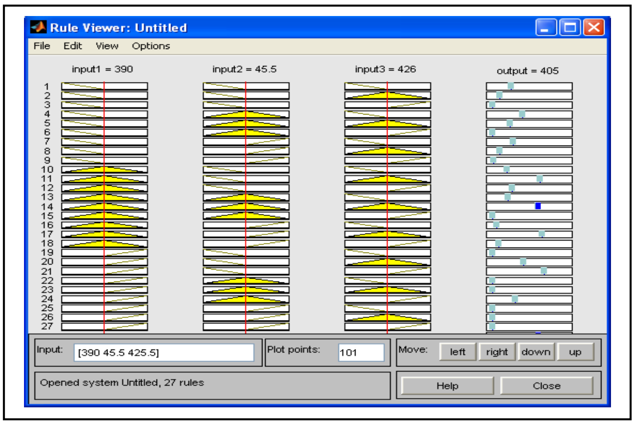

Figure3:Rule Viewer For Kemerer Model, AND Operation In Anfis Editor

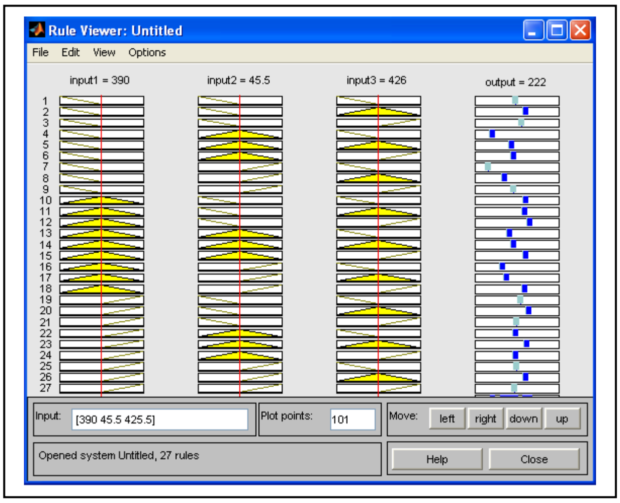

Figure4:Rule Viewer For Kemerer Model, OR Operation In Anfis Editor

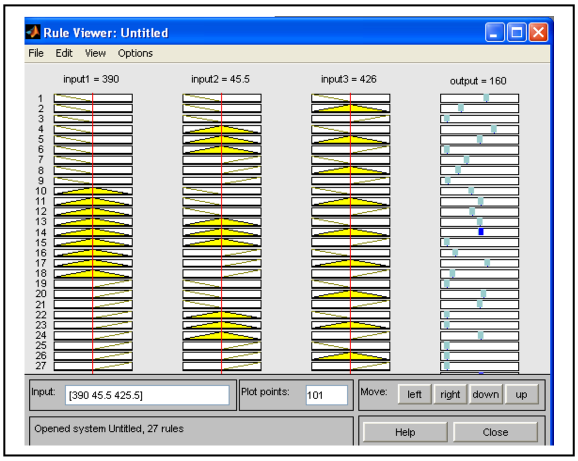

Figure5:Rule Viewer For Small Project Regression Model, AND Operation In Anfis Editor

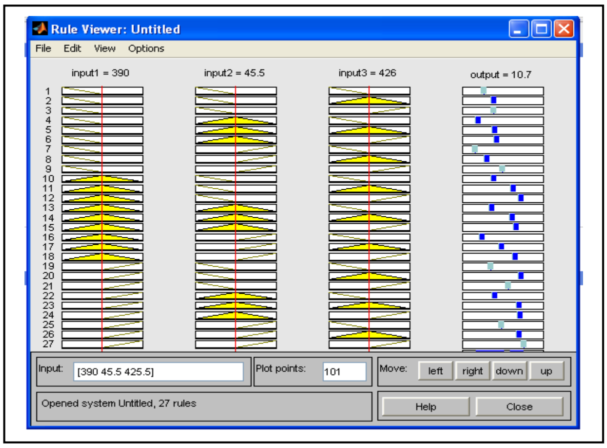

Figure6:Rule Viewer For Albrecht Model, OR Operation In Anfis Editor

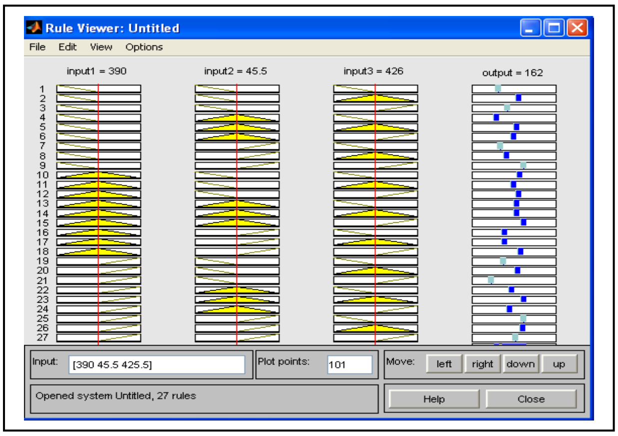

Figure6:Rule Viewer For Small Project Regression Model, OR OperationIn Anfis Editor

In our work we have implemented the FP oriented models Albrecht model,Small project regression model,Kemerer model for both AND and OR logical operations.After implementation we have found out the results as shown in the table1. After performing implementation on 80 projects for Albrecht model,160 projects in Kemerer model and 150 projects in Small Project regression model it is found out that small project regression model is found out to be more effective in terms of it's MMRE and MMER when compared to other function point oriented estimation models.

Shama Kousar Jabeen.A and Mrs. B.Arthi[15] have shown in there work for effort estimation using function points with training the dataset using fuzzy logic approach. Shama Kousar Jabeen.A and Mrs.B.Arthi[14] have shown in there another work for size estimation with neuro fuzzy logic approach for Albrecht dataset.In this paper we have extended of [14] for implementation in other function point(FP) oriented models.

Table1 :Comparison Of Effort Estimation Parameters For FP Oriented Estimation Models Using ANFIS Toolbox

S.NO	FUNCTION POINT ESTIMATION MODEL NAME	MMRE	MMER	MRE	MER
1	ALBRECHT MODEL-AND OPERATION	0.0353	0.0092	2.8213	0.7383

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

2	ALBRECHT MODEL-OR OPERATION	0.0005	0.0006	0.0429	0.0448
3	SMALL PROJECT REGRESSION MODEL-AND OPERATION	0.0001	0.0001	0.017	0.0168
4	SMALL PROJECT REGRESSION MODEL-OR OPERATION	0.00003	0.00002	0.0045	0.0044
5	KEMERER MODEL-AND OPERATION	0.0004	0.0004	0.0591	0.0628
6	KEMERER MODEL-OR OPERATION	0.0045	0.0026	0.7165	0.4174

V. CONCLUSION

We have not implemented our work for cost drivers and other parameters. In this implementation it can be concluded that small prject model is highly effective followed by kemerer model.Future work must consider other machine learning methods with a larger dataset.

ACKNOWLEDGEMENT

I sincerely thank all those who have offered there support and valuable guidelines in accomplishing this work. I also thank my family for there constant support in all my endeavours.

REFERENCES

1. B. Boehm, Software Engineering Economics, Prentice-Hall,City:Englewood Cliffs,State:N.J., 1981.
2. B.Boehm,C.Abts,S.Chulani,"Software development Cost Estimation approaches-a survey",in J.C.Baltzer, Annals Of Software Engineering,Vol.10,Science Publishers,pp.177-205,Issue ¼.
3. International Function Point Users Group (IFPUG) <http://www.ifpug.org/>
4. J. Wong, D. Ho, L.E. Capretz, An investigation of using Neuro-Fuzzy with software size estimation, in: ICSE Workshop on Software Quality, WOSQ'09, 2009, pp. 51–58.
5. K. Srinivasan and D. Fisher, "Machine Learning Approaches to Estimating Software Development Effort", IEEE Transactions on Software Engineering, 21 (2), 1995, pp. 126-136.
6. K.K. Shukla, Neuro-genetic prediction of software development effort,Information & Software Technology 42 (10) (2000) 701–713.
7. L.Putnam and W.Myers,"Measures For Excellence",Yordon Press,1992
8. M.W.Nisar,W.Yong-Ji,M.Elahi,"Software Effort Effort Estimation Using Fuzzy Logic –A survey", In Fifth International Conference on Fuzzy Systsms and Knowledge Discovery,FKSD'08, pp-421-427, 2008
9. Moataz A.Ahamed, Irfan A.Ahmad, Jarallah S.Alghamdi,"Probabilistic Size Proxy For Software Effort prediction: A framework", King Fahd University Of Petroluem and Minerals,Dhahran 31261,Saudi Arabia, pp 241-251, 2013.
10. Moataz A.Ahmad, Zeeshan Muzaffar, "Handling Imprecision and uncertanty in software development effort prediction : A Type-2 fuzzy logic based framework", Journal Of Information and Software Technology(IST),Vol.51,No.3,pp.92-109,http://dx.doi.org/10.1016/j.infsof.2008.09.004.2009
11. R.S.Pressman, Software Engineering,"A Practioner's Approach", Mcgraw Hill, ,pp-674-702,2001.
12. S.H.Kan,Metrics and Models in Software Quality Engineering,Second Edition,Pearson Education Inc,2002.
13. Sandeep Kad,Vinay Chopra,"Software development Effort Estimation using Soft Computing",International Journal Of Machine Learning and Computing (IJMLC),V2.186,Vol.2, No.5,pp.548-551, ISSN: 2010-3700,DOI:10.7763,October 2012.
14. Shama Kousar Jabeen.A and Mrs.B.Arthi,"A Proposal On Size Metric Framework Modelling For Software Effort Estimation Models Using Neuro Fuzzy Logic Approach"Proceedings of fourth International Conference On "Advance Computing,Control Systems,Machnies and Embedded Technology",(ICACT),pp-1372-1376,ISBN:978-93-80757-74-2.
15. Shama Kousar Jabeen.A and Mrs.B.Arthi,"Size Proxy Metric Framework Modeling Of Software Effort Estimation in Soft Computing", Journal Of Emerging Technolgies,Vol:8,pp:1-5,Special Issue-III,Feb 2014.
16. Visual Use Case Tool, <http://www.technosolutions.com/topteam-use-case.html>.