Hybrid Neuro-Fuzzy Systems for Software Development Effort Estimation

Rama Sree P
Dept. of Computer Science & Engineering, Aditya Engineering College
Jawaharlal Nehru Technological University Kakinada
Kakinada, Andhra Pradesh, India
ramasree_p@rediffmail.com

Prasad Reddy P. V. G. D
Dept. of Computer Science and Systems Engineering
Andhra University
Visakhapatnam, Andhra Pradesh, India
prasadreddy.vizag@gmail.com

Sudha K. R
Dept. of Electrical Engineering
Andhra University
Visakhapatnam, Andhra Pradesh, India
arsudhaa@gmail.com

***Abstract*- The major prevailing challenges for Software Projects are Software Estimations like cost estimation, effort estimation, quality estimation and risk analysis. Though there are several algorithmic cost estimation models in practice, each model has its own pros and cons for estimation. There is still a need to find a model that gives accurate estimates. This paper is an attempt to experiment different types of Neuro-Fuzzy Models. Using the types of Neuro-Fuzzy Models for software effort prediction is a relatively unexplored area. Two case studies are used for this purpose. The first is based on NASA-93 dataset and the other is based on Maxwell-62 dataset. The case studies were analyzed using six different criterions like Variance Accounted For (VAF), Mean Absolute Relative Error (MARE), Variance Absolute Relative Error (VARE), Mean Balance Relative Error (Mean BRE), Mean Magnitude Relative Error (MMRE) and Prediction. From the results and from reasoning, it is concluded that Type B-Compensation Neuro-Fuzzy Model with more fuzzy rules is best suitable for cases in which the datapoints are more linear. Type J Neuro-Fuzzy Model with more fuzzy rules is best suitable for cases in which the datapoints are not linear.**

Keywords-Effort Estimation, Fuzzy Logic, Neural Nets, Neuro-Fuzzy Models, NASA-93 Dataset, Maxwell-62 Dataset.

I. INTRODUCTION

Initially algorithmic cost estimation models were used to predict costs and efforts for software projects. In these models, mathematical formulae derived based on some historical data were used [1]. But, the results produced by these statistical cost estimation models were poor. This is because project data, available in the initial stages of project is often incomplete, inconsistent, uncertain and unclear. The need for accurate effort prediction in software project management is an ongoing challenge. Soft Computing approaches like Fuzzy Logic [2] [3] & Neural Networks [4] [5], Neuro-Fuzzy Systems are more apt in case of such predictions. This paper is an attempt in using the Neuro-Fuzzy Models for estimating the effort of software projects. The present paper uses NASA-93 dataset and Maxwell-62 dataset.

A. *NASA-93 Dataset*

The NASA-93 dataset [6] depends on Intermediate COCOMO proposed by Barry Boehm in 1981. Intermediate COCOMO Development Effort (DE) Calculation [7] is shown in Table I. DE is measured in man-months or person/months.

TABLE I. DE FOR THE INTERMEDIATE COCOMO

Development Mode	**Intermediate Effort Equation**
Organic	DE = EAF * 3.2 * (SIZE)$^{1.05}$
Semi-detached	DE = EAF * 3.0 * (SIZE)$^{1.12}$
Embedded	DE = EAF * 2.8 * (SIZE)$^{1.2}$

The NASA-93 dataset uses factors as shown in the Table II.

TABLE II. NASA-93 PROJECT DATA FACTORS

S. No.	Factors	S. No	Factors
1	Mode	10	Computer Turnaround Time
2	Size	11	Analyst Capability
3	Actual Effort	12	Applications Experience
4	Required Software Reliability	13	Programmer Capability
5	Database Size	14	Virtual Machine Experience
6	Product Complexity	15	Language Experience
7	Execution Time	16	Modern Programming
8	Main Storage Constraint	17	Use Of Software Tools
9	Virtual Machine Volatility	18	Required Development Schedule Size

B. *Maxwell-62 Dataset*

The Maxwell-62 dataset [8] uses the factors shown in Table III for software effort prediction. The dataset includes 62 records. This dataset is mainly used for 2 reasons. Firstly, to show that the proposed Neuro-Fuzzy Models gives accurate results even with the latest software development factors when compared with the old COCOMO factors that is developed in 1981. Secondly, to show that the proposed Neuro-Fuzzy Models gives accurate results even when there is no analytical equation to calculate the effort depending on the factors as like in COCOMO. Maxwell-62 dataset is the 62 projects data collected from one of the biggest commercial banks in Finland.

TABLE III. MAXWELL-62 PROJECT DATA FACTORS

S. No.	Factors	S. No	Factors
1	Application type	14	Software complexity
2	Hardware platform	15	Requirements volatility
3	Database	16	Quality Requirements
4	Interface	17	Efficiency Requirements
5	Source	18	Installation Requirements
6	Telonuse	19	Staff Analysis Skills
7	No. of languages used	20	Staff application knowledge
8	Customer participation	21	Staff tool skills
9	Development environment	22	Staff team skills
10	Staff availability	23	Duration
11	Standards Use	24	Size
12	Methods Use	25	Effort
13	Tools Use		

II. SOFT COMPUTING APPROACHES

In Soft Computing approaches, the system effectively "learns" how to estimate from a training set of completed projects. Soft computing is a consortium of methodologies centering in Fuzzy Logic, Neural Networks and Neuro Fuzzy Inference Systems [9]. Soft Computing approaches aims at solving problems

- Where there is no analytical equation between inputs & outputs.
- Where non linear relationship exists between inputs & outputs.

In this section the focus is on Fuzzy Logic, Neural Networks and then the proposed Neuro-Fuzzy Models.

A. Fuzzy Logic

When the systems are not suitable for analysis by conventional approach or when the available data is uncertain, inaccurate or vague, a fuzzy model is used [10] [11]. The Fuzzy logic maps an input space to an output space using a list of if-then statements called rules. All rules are evaluated in parallel not bothering the order of the rules [12] [13] [14].

Advantages:

1) The main advantage of using the fuzzy ranges is that it predicts the effort for projects that do not come under a precise mode i.e. comes in between 2 modes, where the situation cannot be handled using the COCOMO.
2) Fuzzy logic is tolerant of imprecise data.
3) Fuzzy logic is based on natural language.

Disadvantages:

1) As the whole work has to be redefined for a newer dataset it is hard to maintain a degree of meaningfulness
2) As the answers are confined to what is written in its rule base, it is incapable to generalize.
3) Demands the presence of an expert to write the rules.

B. Neural Networks

Neural network [15][16], a massive parallel distributed processor made up of simple processing units which has a natural propensity for storing experimental knowledge and making it available for use, resembles the brain in two respects[17][18]:

1) Knowledge is acquired by the network from its environment through a learning process.
2) Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

Advantages:

1) Artificial neural networks is very useful in problems where there is a complex relationship between inputs and outputs as it can model complex non-linear relationships and approximate any measurable function.
2) Many different algorithms are available to choose from.

Disadvantages:

1) There is no clear guidance on how to design neural nets like for e.g. how many hidden layers are to be present.
2) Accuracy depends on larger training dataset which is not always available.
3) They are effectively black boxes- once given the inputs; the generated outputs have to be accepted.

C. Neuro-Fuzzy Model

While neural networks are good at recognizing patterns, they are not good at explaining how they reach their decisions. Fuzzy logic systems, which can reason with imprecise information, are good at explaining their decisions but they cannot automatically acquire the rules they use to make those decisions [19]. These limitations have been a central driving force behind the creation of intelligent hybrid systems where two or more techniques are combined in a manner that overcomes the limitations of individual techniques [20]. The hybridization of neural networks and fuzzy logic, the basic idea behind the Neuro Fuzzy system can be done in many ways [21]. The typical Neuro-Fuzzy Models are presented in the next section [22].

D. Typical Neuro-Fuzzy Models

In this section a brief description about the different types of Neuro-Fuzzy Models from Type A to Type K is presented. The present paper experiments the Type A, B (Unified & Compensation), G and J Neuro-Fuzzy models as they are feasible to the software effort estimation and follows the strategy of tuning the Fuzzy Rules using neural networks.

Type A

Suppose that a system has two functions of fuzzy rules and ANN independently. The fuzzy rules handle some input and output variables while ANN does the others (Figure 1). It can be seen that the fuzzy rules deal with the different input variables from those of ANN. The model is referred to as Type A.

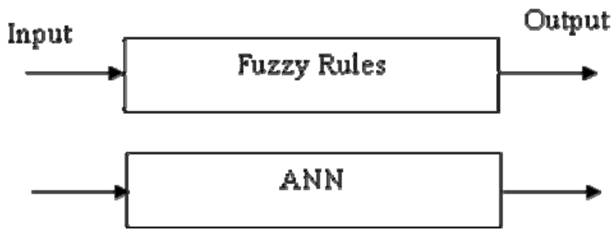

Figure 1. Neuro Fuzzy Type A

Type B

Fuzzy rules and ANN may be placed in parallel. The model is called Type B. Depending on the role of FL and ANN, it may be divided into the following models:

1) Unified model (see Figure 2)
2) Compensation model (see Figure 3)

In the Unified model, information processing is equally done for FL and ANN. Also, FL compensates the results obtained by ANN in the compensation model, and vice versa.

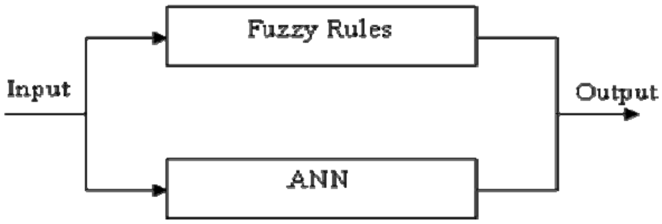

Figure 2. Neuro Fuzzy Type B – Unified model

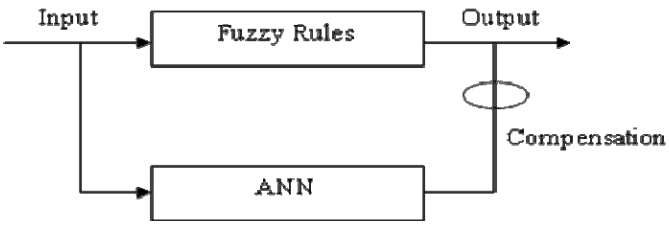

Figure 3. Neuro Fuzzy Type B – Compensation model

Type C

ANN and Fuzzy Logic Controller may be placed in series as shown in Figure 4. so that two-phase inference is possible. In Type C, ANN is used to adjust the fuzzy rules to evaluate output variables.

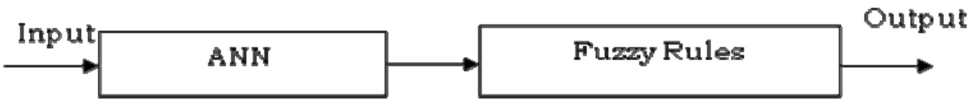

Figure 4. Neuro Fuzzy Type C

Type D

Fuzzy Logic Controller and ANN may be placed in series as shown in Figure 5, so that two-phase inference is possible. In Type D, FLC is used to adjust the ANN parameters to evaluate output variables.

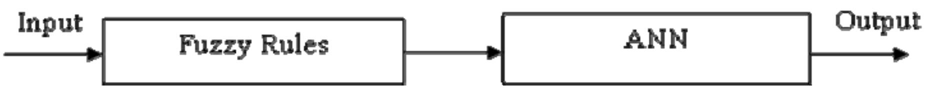

Figure 5. Neuro Fuzzy Type D

Type E

Figure 6 shows Type E of the Neuro fuzzy model. A fuzzy model is used to handle fuzzy rules in which the goal and parameters of the fuzzy control are evaluated. It should be noted that ANN contributes to determination of constructing the fuzzy rules. In other words, ANN plays an assistant role in the fuzzy model.

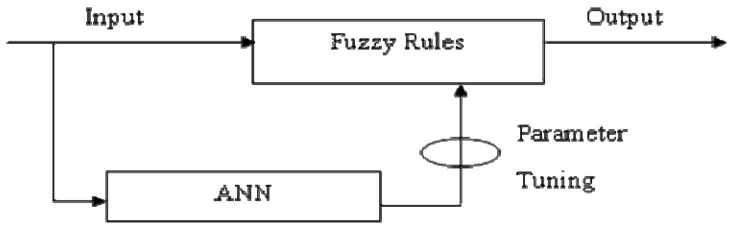

Figure 6. Neuro Fuzzy Type E

Type F

Figure 7 shows Type F of the Neuro-Fuzzy model. ANN model is used to handle fuzzy rules in which the goal and parameters of the fuzzy control are evaluated. It should be noted that FLC contributes to determination of constructing the NN weights. In other words, FLC plays an assistant role. It is called as Fuzzy-Neuro inference system.

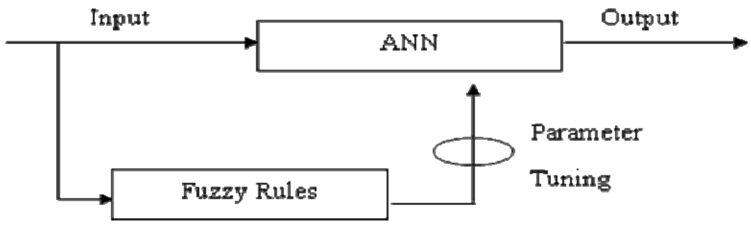

Figure 7. Neuro Fuzzy Type F

Type G

Type G makes use of the integration of fuzzy rules and ANN so that the supervised learning of ANN is used to evaluate the membership function shape and the weight of true value of fuzzy rules (see Figure 8). As a learning scheme, the steepest decent method is used the error back propagation algorithm of the multilayer perceptron. The difference between Types E and G is that only the function of the ANN learning is used in Type G to tune up fuzzy rules to improve the solution accuracy.

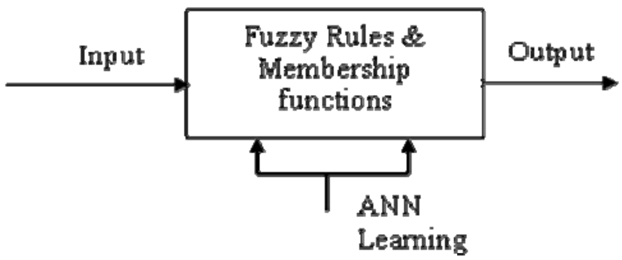

Figure 8. Neuro Fuzzy Type G

Type H

The model of Type H has function that fuzzy rules of if-then are expressed with the ANN construction. The model is useful in a sense that the computation process of fuzzy inference may be represented by a learning model. This concept is shown in Figure 9. Since the ANN represents the fuzzy rules, the output variable after the ANN learning corresponds to the inference value of fuzzy model.

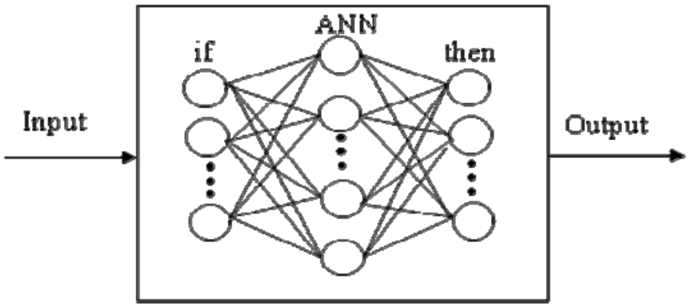

Figure 9. Neuro Fuzzy Type H

Type I

A fuzzy inference model is identified with ANN to clarify the relationship between the premise and consequence of fuzzy rules as shown in Figure 10. The ANN model is constructed after fuzzy sets of the premise and consequence are assigned to input and output of the learning data of ANN, respectively. As a result, the input and output variables of the model correspond to the value of the fuzzy membership functions. Specifically, studies on ANNs representing fuzzy rules and fuzzy operators have been done.

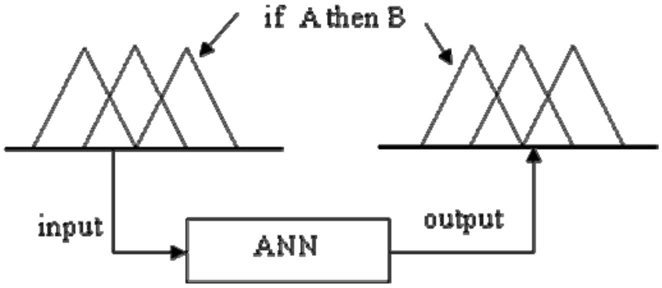

Figure 10. Neuro Fuzzy Type I

Type J

A part of fuzzy rules in the fuzzy model is expressed by ANNs in Type J (see Figure 11). The ANN model is used to substitute for some fuzzy rules so that the errors of the fuzzy membership functions or the consequence are reduced. The difference between Types G and J is that ANNs becomes a subsystem of fuzzy rules in Type J.

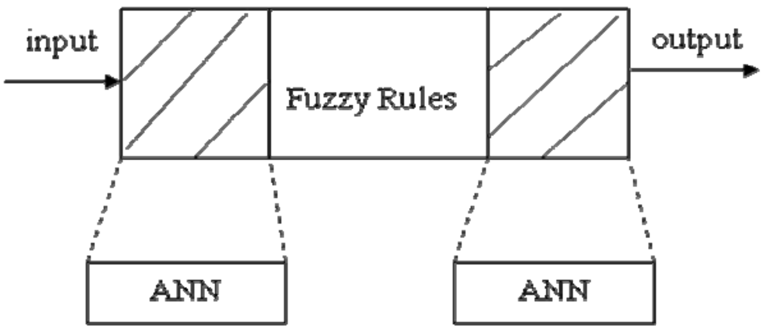

Figure 11. Neuro Fuzzy Type J

Type K

The model corresponds to a generalized Neuro Fuzzy model (see Figure12). It is a kind of an extension of ANN in a sense that the weights between neurons are fuzzified. That implies that it can handle input data as a fuzzy number. It is necessary to develop more sophisticated learning algorithms in consideration of fuzzy logic.

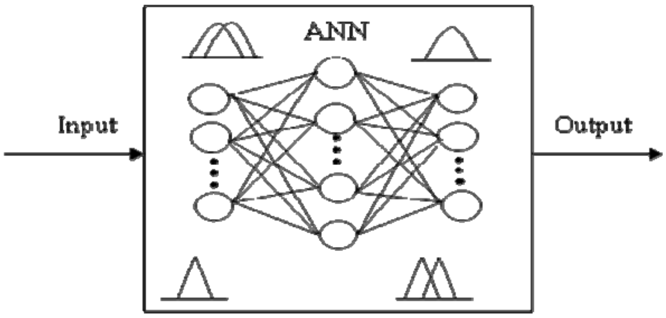

Figure 12. Neuro Fuzzy Type K

III. VARIOUS CRITERIONS FOR ASSESSMENT OF ESTIMATION MODELS

1. Variance Accounted For (VAF)

$$VAF\ (\%) = \left(1 - \frac{\operatorname{var}\left(E - \hat{E}\right)}{\operatorname{var}\ E}\right) \times 100 \qquad (1)$$

2. Mean Absolute Relative Error (MARE)

$$MARE\ (\%) = \frac{\sum f\,(R_E)}{\sum f} \times 100 \qquad (2)$$

3. Variance Absolute Relative Error (VARE)

$$VARE\,(\%) = \frac{\sum f\,(R_E - meanR_E)}{\sum f} \times 100 \qquad (3)$$

4. Prediction (n)

Prediction at level n is defined as the % of projects that have absolute relative error less than n.

5. Balance Relative Error (BRE)

$$BRE = \frac{\left|E - \hat{E}\right|}{\min(E, \hat{E})} \qquad (4)$$

E = actual effort $\quad \hat{E}$ = estimated effort

Absolute Relative Error (RE) = $\left|\frac{\hat{E} - E}{\hat{E}}\right|$ (5)

6. Mean Magnitude Relative Error (MMRE)

$$MMRE(\%) = \frac{1}{N}\sum_{i=1}^{N} MRE_i \times 100 \qquad (6)$$

Where $MRE = \left|\frac{\hat{E} - E}{\hat{E}}\right|$, N = No. of Projects

E = estimated effort $\quad$ Ê = actual effort

A model which gives higher VAF, Pred(30) and lower VARE, BRE, MARE and MMRE is considered to be better than the other models [23][4].

IV. EXPERIMENTAL STUDY

Type A, Type B-Unified, Type B-Compensation, Type G and Type J Neuro-Fuzzy Models are experimented using the NASA-93 dataset and Maxwell-62 dataset. The number of datasets passed to Fuzzy Logic Inference System and Neural Networks as training data and check data are shown in Table IV. The number of fuzzy rules framed is also shown in Table IV. Table V shows the comparisons of various Neuro-Fuzzy Models against

NASA-93 dataset and Maxwell-62 dataset. For Type J in NASA-93, 6 ANN's are used to substitute Fuzzy Rules and in Maxwell-62, 3 ANN's are used to substitute Fuzzy Rules. The output of all these Neuro-Fuzzy models is the Effort calculated in man-months (mm) or person-months (pm).

TABLE IV. TRAINING DATA & CHECK DATA FOR FUZZY LOGIC & NEURAL NETWORKS

S. No	Neuro-Fuzzy Models	Fuzzy Logic Inference System			Neural Networks	
		No. of Train datasets	No. of Check datasets	No. of Fuzzy Rules	No. of Train datasets	No. of Check datasets
Case Study 1: NASA-93 DATASET	Type A Model	30	35	1	49	58
	Type B-Unified Model	43	46	1	43	47
	Type B- Compensation	83	93	6	83	93
	Type G Model	83	93	3	83	93
	Type J Model	84	93	33	84	93
Case Study 2: Maxwell-62 DATASET	Type A Model	17	20	17	35	42
	Type B-Unified Model	26	31	26	26	31
	Type B- Compensation	52	62	51	52	62
	Type G Model	52	62	51	52	62
	Type J Model	52	62	42	52	62

TABLE V. COMPARISON OF VARIOUS NEURO-FUZZY MODELS

NASA 93 DATASET						
Neuro- Fuzzy Models	***VAF***	***MARE***	***VARE***	***BRE***	***MMRE***	***Pred(30)%***
Type A Model	95.2414	14.1332	83.4659	0.7342	100.5496	77.4194
Type B –Unified Model	95.9943	79.686	3.49E+03	0.9261	33.0362	75.2688
Type B –Compensation Model	99.9168	5.2765	3.2397	0.1824	17.7746	93.5484
Type G Model	96.1462	14.1432	130.0434	0.6639	118.2494	60.2151
Type J Model	99.8987	4.6804	1.7966	0.1021	10.1583	94.6237
MAXWEL-62 DATASET						
Neuro- Fuzzy Models	***VAF***	***MARE***	***VARE***	***BRE***	***MMRE***	***Pred(30)%***
Type A Model	98.3517	15.7994	25.7884	0.1895	10.5587	87.0968
Type B –Unified Model	97.3388	6.4574	2.9073	0.2339	23.2965	91.9355
Type B –Compensation Model	99.7997	2.8781	5.0527	0.0288	1.0336	98.3871
Type G Model	99.2496	8.1346	7.1013	0.0867	5.324	90.3226
Type J Model	99.2999	5.8828	6.4331	0.0779	5.383	93.5484

V. CONCLUSION

Referring to Table V, it is clear that Neuro Fuzzy Type B-Compensation Model and Neuro Fuzzy Type J Model yields better results for maximum criterions when compared with the other models. As per results based on VAF, MARE, VARE, Mean BRE, MMRE & Pred (30), Type J Model is best suitable for NASA-93 dataset and Type B-Compensation Model is best suitable for Maxwell-62 dataset. The suitable Neuro-Fuzzy Model for a dataset depends on two factors.

1) Linearity in the dataset
2) No. of Fuzzy Rules

If the dataset has more linear datapoints, it is easier to estimate. The Neuro-Fuzzy Model which uses more number of rules would give accurate results as almost all the combinations of inputs are being represented as rules for the fuzzy inference system.

For Nasa-93 dataset, the datapoints are not spread uniformly across the output range. The output range is varying from 8.4 pm to 8211 pm. Nearly 1/3rd of dataset(31 records) have output effort less than 100 pm(part 1/80th of Max effort) . 50% of dataset (46 records) have effort less than 225 pm (part 1/36th of max effort). 90% of dataset (83 records) have effort less than 1000pm (part 1/8th of max effort). This clearly shows that the datapoints are not linear. Therefore, we need to frame a Neuro-Fuzzy Model that has more number of fuzzy rules and use more learning schemes to tune different sets of fuzzy rules for solution accuracy. The number of fuzzy rules for Type J model is 33 and 6 ANN's are used to substitute fuzzy rules (see Table IV). For all these reasons and from experimentation (see Table V) we conclude that, Type J Neuro-Fuzzy Model is best than the other models for Nasa-93 dataset.

For Maxwell-62 dataset [24], the datapoints are spread uniformly across the output range. The output range is varying from 583 pm to 63694 pm. Nearly 1/3rd of records (21 records) have output effort less than 3500 pm (part 1/18th of max effort). 50% of dataset (31 records) have effort less than 5100 pm (part 1/12th of max effort). 90% of dataset (56records) have effort less than 15000pm (part 1/4th of max effort). This clearly shows that the datapoints are more linear. Because the datapoints are more linear, we need not use more number of learning schemes to tune the Neuro-Fuzzy Model. We need to frame a Neuro-Fuzzy Model which uses more number of Fuzzy rules. The numbers of fuzzy rules used for Type B-Compensation and Type G are same, but Type B-Compensation Neuro-Fuzzy Model gives better results as Fuzzy outputs compensate Neural Networks outputs and vice versa. For all these reasons and from experimentation (see Table V) we conclude that, Type B-Compensation Neuro-Fuzzy Model is best than the other models for Maxwell-62 dataset.

Finally, we can conclude that, Type J Neuro-Fuzzy Model is best when the number of fuzzy rules used is more and datapoints are not linear. Type B-Compensation Neuro-Fuzzy Model is best when number of fuzzy rules used is more and datapoints are more linear.

REFERENCES

[1] Jainendra K. Navlakha, "Choosing a Software Cost Estimation Model for Your Organization: A Case Studv ", Elsevier Science Publishers B.V. (North-Holland), 1990,255-261

[2] Prasad Reddy P.V.G.D, Sudha K.R, Rama Sree P, & Ramesh S, (2010). Fuzzy Based Approach for Predicting Software Development Effort. International Journal of Software Engineering, Vol 1, Issue 1, 1-11.

[3] Prasad Reddy P.V.G.D, Sudha K.R, Rama Sree P, "Application of Fuzzy Logic Approach to Software Effort Estimation", International Journal of Advanced Computer Science & Applications, Vol 2, Issue 5, May 2011, 87-92.

[4] Prasad Reddy P.V.G.D, Sudha K.R , Rama Sree P., Ramesh S, (2010). Software Effort Estimation using Radial Basis and Generalized Regression Neural Networks. Journal of Computing, Vol 2, Issue 5, 87-92.

[5] Prasad Reddy P.V.G.D, Sudha K.R, Rama Sree P, " Prediction of Software Development Effort Using RBNN And GRNN", International Journal of Computer Science Engineering and Technology, Vol 1, Issue 4, May 2011, 185-190.

[6] NASA 93 DatasetSource: http://promisedata.org/repository/data/nasa93/nasa93.arff, Jairus Hihn, JPL, NASA, Manager SQIP Measurement &Benchmarking Element, Feb 8 2006

[7] B.W. Boehm. (1981). Software Engineering Economics. Prentice-Hall, Englewood Cli4s, NJ.

[8] Maxwell DatasetSource: http://promisedata.org/repository/data/maxwell/maxwell.arff, Creator :Dr. Katrina D. Maxwell, Donor: Yanfu Li (liyanfu@nus.edu.sg), March 21 ,2009

[9] Dr. Arvinder Kaur, Kamaldeep Kaur, & Dr. Ruchika Malhotra, (2010). Soft Computing Approaches for Prediction of Software Maintenance Effort. International Journal of Computer Applications,Vol.1, No. 16.

[10] Harish Mittal, & Pradeep Bhatia, (2007). Optimization Criteria for Effort Estimation using Fuzzy Technique. CLEI Electronic Journal, Vol10, No 1, Paper 2.

[11] R. Babuska, Fuzzy Modeling For Control, Kluwer Academic Publishers, Dordrecht, 1999

[12] Moshood Omolade Saliu, Adaptive Fuzzy Logic Based Framework for Software Development Effort Prediction, King Fahd University of Petroleum & Minerals, April 2003

[13] Iman Attarzadeh and Siew Hock Ow, Software Development Effort Estimation Based on a New Fuzzy Logic Model, IJCTE, Vol. 1, No. 4, October2009

[14] Xishi Huang, Danny Ho,Jing Ren, Luiz F. Capretz, A soft computing framework for software effort estimation, Springer link, Vol 10, No 2 Jan-2006

[15] A. Idri, T. M. Khoshgoftaar, & A. Abran, (2000). Can neural networks be easily interpreted in software cost estimation. IEEE Trans, Software Engineering, Vol. 2, pp. 1162 – 1167

[16] Alain Abran, Ali Idri, & Samir Mbarki, (2004). Validating and Understanding Software Cost Estimation Models based on Neural Networks. IEEE, 0-7803-8482-2/04.

[17] LiMin Fu. (2003). Neural Networks in Computer Intelligence. Tata McGraw-Hill Edition pp.94-97.

[18] Dawson, C.W. (1996). A neural network approach to software projects effort estimation. Transaction: Information and Communication Technologies, Volume 16, pages 9.

[19] A. Kablan (2009). Adaptive Neuro-Fuzzy Inference System forFinancial Trading using Intraday Seasonality Observation Model. World Academy of Science, Engineering and Technology 58.

[20] R. Jang, (1993). ANFIS: Adaptive network-based fuzzy inference system. IEEETransactions on Systems, Man and Cybernetics, Vol. 23 (3) pp 665-685.

[21] Xishi Huang, Danny Ho, Jing Ren, & Luiz F. Capretz, (2007). Improving the COCOMO model using a neuro-fuzzy approach. Science Direct, Applied Soft Computing 7, 29–40.

[22] Tutorial on Fuzzy Logic Applications in Power Systems, IEEE - PES Winter Meeting in Singapore, January, 2000

[23] Zhiwei Xu, Taghi M. Khoshgoftaar, Identification of fuzzy models of software cost estimation, Fuzzy Sets and Systems 145 (2004) 141–163

[24] G. Boetticher, T. Menzies and T. Ostrand, PROMISE Repository of empirical software engineering data http://promisedata.org/repository, West Virginia University, Department of Computer Science, 2007