Performance analysis of FCM based ANFIS and ELMAN neural network in software effort estimation

Praynlin[1] and Latha[2]
[1]Department of Computer science, Engineering, Noorul Islam University, India
[2]Department of computer science, Engineering from Bharathiar university, India

Abstract: *One of the major challenges confronted in the software industry is the software cost estimation. It is very much related to, the decision making in an organization to bid, plan and budget the system that is to be developed. The basic parameter in the software cost estimation is the development effort. It tend to be less accurate when computed manually. This is because, the requirements are not specified accurately at the earlier stage of the project. So several methods were developed to estimate the development effort such as regression, iteration etc. In this paper a soft computing based approach is introduced to estimate the development effort. The methodology involves an Adaptive Neuro Fuzzy Inference System (ANFIS) using the Fuzzy C Means clustering (FCM) and Subtractive clustering (SC) technique to compute the software effort. The methodology is compared with the effort estimated using an Elman neural network. The performance characteristics of the ANFIS based FCM and SC are verified using evaluation parameters.*

Keywords: *Software development, cost, effort estimation, process planning, ANFIS.*

Received January 4, 2014; accept July 9, 2014

1. Introduction

Software effort estimation has become a major challenge over the past few decades in the software development industry [3]. Effort estimation deals with estimating the required effort based on some parameters like Function point, Lines of code etc., and it is necessary, to effectively manage the entire software development process. Effort estimation at the earlier stage is less accurate [17,19, 26]. This is due to the difficulty in predicting the amount of time, resource, and investment required for the project. Cost, effort, and schedule are the three interrelated factors. It plays a significant role in the process planning. Most part of the cost in developing a software is directly related to the required human effort. Other expenses related to infrastructure are very low or negligible. Thus the expense associated with the human resource, and their wages directly influence the cost of the project. The effort required is thus known. Schedule is calculated, using the effort and the required working hours of the employee. Thus, effort evolves as the most basic parameter in the software development.

The software cost estimation process can be classified into two methods namely algorithmic and non-algorithmic method. Algorithmic methods are formula based approaches like Constructive Cost Model (COCOMO) and Software Life cycle management (SLIM). They are the most widely used methods. Barry Boehm put forward COCOMO and SLIM is put forward by Putnam. The non-algorithmic method include estimation methods like Expert judgment, Regression models, Halstead model, Walston-felix mode, all machine learning methods etc., [1]. Due to the emergence of soft computing in the last decade, contribution of neural network to estimate the effort goes on increasing rapidly [16]. Since neural networks is good in learning and fuzzy logic is good in dealing with uncertainty, the hybrid approach ANFIS is considered in this paper. The paper is organized as follows: Section 2 discusses about some of the related works existing in the literature. Section 3 deals with the software effort estimation and ANFIS. The dataset description is provided in section 4 and the experimental setup is shown in section 5. Section 6 discusses about the various evaluation parameters in the software effort estimation. Section 7 gives a brief description about the results of the paper and the paper concludes with section 8.

2. Literary Review

There are so many methods to estimate the cost of the project. They are analogy-based methods [28, 2], Bayesian methods [5, 7] and Regression methods [10, 18]. Improved version of regression like adaptive regression can also be used to estimate the effort [31]. The Artificial Neural Networks used to estimate the effort is mostly based on comparing the accuracy of the algorithmic model than considering the suitability of the approach for developing software effort prediction systems. Witting and Finnie [29, 30] described the use of back propagation algorithm on a multilayer perceptron to predict the software development effort. Several other neural network architecture are used to estimate the effort. Ali Idri [13] used radial basis function network to estimate the software development effort. The general regression neural network can also be used to estimate the effort. Statistical tests are done to validate the performance of the network [8] Prasad Reddy [24] compared the difference between radial basis function network and ceneral regression neural network.

With the arrival of fuzzy logic in 1965 by Lotfizadeh a new hybrid approach combining the neural network and fuzzy logic techniques begin to develop. Jyh– shing and Roger Jang [14] proposed a new method called Adaptive-Network-Based Fuzzy Inference system. A typical Constructive cost model COCOMO based on neuro-fuzzy concept was proposed by Xishi Huang *et.al.* [12]. Venus Marza et.al [20] claims that the neuro-fuzzy combination has better estimation capabilities than the standalone neural network and fuzzy logic concept. Their results show that the Mean Magnitude of Relative Error (MMRE) of a neuro-fuzzy system is low, compared to the MMREs of neural network or a fuzzy logic system itself. Sometimes, a hybrid neuro-genetic approach is also used to estimate the software development effort [27]. Grey relation analysis and regression method has been used to predict the effort by Nagpal Geeta [9].

3. Effort Estimation of a Software

In the software industry, project managers more frequently estimate the effort required for software development. Using the estimated effort, the cost and duration required for the development cycle are computer. Planning the software development cycle, monitoring and controlling the progress can be accomplished effectively, knowing the effort required accurately [22]. Thus, software effort plays a crucial role in the software industry.

The effort estimation of the software can be done by relating similar tasks, which have already been developed. Estimation using similar tasks has an inexact nature, as it depends on numerous hazy factors. Software scheduling and cost estimation helps in effectively planning and tracking the software project.

Some of the difficulties in effort estimation is that, it is difficult, to clearly understand the requirements at the initial phase of the development. Some may be expert in one programming language whereas another may have expertise in some other programming language. For some programming languages, there will be sophisticated development environment like GUI and others may not have it [21].

4. ELMAN and ANFIS in Effort Estimation

4.1. ELMAN Neural Network

Jeffrey L. Elman in 1990 proposed the Elman network for a complete estimation model. The Elman network is a feed forward neural network with an input layer, hidden layer, output layer and a special layer called context layer. The output of each hidden neuron is copied into a specific neuron in the context layer.

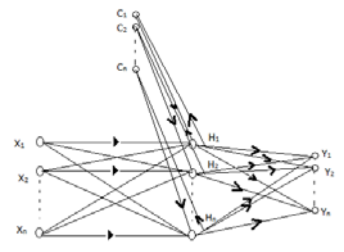

Figure 1. ElMAN network.

A recurrent network is one in which there is a feedback from neuron's output to its input. Let x_1, x_2, x_3----x_n be the input to the network and the output of the network be y_1, y_2, y_3---y_n. The output of the hidden layer (h_1, h_2, h_3--- h_n) are fed back again to hidden layer neuron using the context node (c_1, c_2, c_3----c_n). Unlike feed forward neural networks, recurrent neural networks can use their internal memory to process arbitrary sequences of inputs. The value of the context neuron is used as an extra input signal for all the neurons in the hidden layer one time step later. The weights from the hidden layer to the context layer are set to one and are fixed because the values of the context neurons have to be copied exactly. Furthermore, the initial output weights of the context neurons are equal to half the output range of the other neurons in the network. The network can be trained with gradient descent, back propagation and other optimization methods.

4.2. ANFIS

ANFIS is a hybrid system of neural networks and fuzzy logic concept. Neural network is good in learning and highly interpretable and can function better than regression methods [23] whereas fuzzy logic is good in handling imprecision. The advantage of hybrid neuro-fuzzy approach is that, there is a reduction in training time due to its smaller dimensions, and the networks' capability of initializing the parameters relating to the problem domain [14]. The architecture of ANFIS is a five-layered structure, and it works as a feed-forward neural network. A specific approach in neuro-fuzzy development is that it has shown significant results in modelling nonlinear functions [14].

ANFIS used in this paper is of Takagi sugeno type fuzzy inference system. It applies the combination of least square method and the back propagation gradient descent method for training FIS membership function parameters to emulate the given training dataset [17].

Figure 2 shows a typical ANFIS architecture. The square represents the nodes, that are adaptive and the circle represents the nodes that are fixed. *x* and *y* are the inputs to the node and *f* is the output of the network.

The rule base that governs the fuzzy inference system is given by:

- Rule1: If x is A_1 and y is B_1, then $f_1 = P_1x + q_1y + r_1$
- Rule2: If x is A_2 and y is B_2, then $f_2 = P_2x + q_2y + r_2$

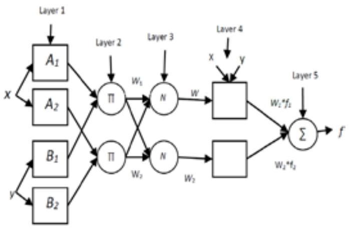

Figure 2. Architecture of ANFIS.

Let x and y be the inputs to the nodes in the first layer with labels A_i or B_{i-2}. The degree to which the input satisfies the fuzzy quantifier A_i or B_{i-2} is given by the membership function.

$$M_{1,i} = \zeta_{A_i}(x) \text{ for i=1, 2} \qquad (1)$$

$$M_{1,i} = \zeta_{B_{i-2}}(y) \text{ for i=3, 4} \qquad (2)$$

$\zeta_A(x)$ is a bell shaped membership function of A_i for a parameter set $\{a_i, b_i, c_i\}$ given by:

$$\zeta_A(x) = \frac{1}{1 + \left|\frac{x - c_i}{a_i}\right|^{2b}} \qquad (3)$$

The layer two performs a multiplication operation on the incoming signals and the output denotes the firing strength of the rule associated with the nodes. The output product is given by:

$$\Psi_{2,i} = w_i = \zeta_{A_i}(x) \times \zeta_{B_{i-2}}(y) \text{ for i=1, 2} \qquad (4)$$

In the third layer the firing strength of the ith rule is normalized with respect to the sum of the firing strengths of all the rules in that node. The normalized firing strength of the third layer is given by

$$\Psi_{3.i} = \overline{w}_\iota = \frac{w_i}{w_1 + w_2} \qquad (5)$$

The outputs of the each node in the fourth layer is a product of the normalized firing strength of the particular node and the rule function f_i affected by the parameter set $\{p_i, q_i, r_i\}$.

$$\Psi_{4,i} = \overline{w}_\iota f_i = \overline{w}_\iota (p_i x + q_i y + r_i) \qquad (6)$$

The fifth layer produces a summation of all the signals from the previous layer outputs given by:

$$\text{Overall output} = \Psi_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i \overline{w}_\iota f_i}{\sum_i w_i} \qquad (7)$$

Chiu [6] put forward the subtractive clustering technique in which data points are considered as the candidates for cluster centres. In subtractive clustering the optimal data point is calculated using the density of the neighbouring data points to mark out a cluster centre. Computation using this method is independent of the problem aspect and has a direct relationship with the number of data points.

Fuzzy C means clustering (FCM), also known as fuzzy ISODATA, is an algorithm used for clustering of data points. In FCM, the degree to which a data point belongs to a cluster is quantified by the membership grade [15]. Bezdek proposed this algorithm in 1973 as an improvement over earlier hard C means (HCM) clustering.

In FCM, the iterations can be carried out after initializing the cluster centres. However, convergence to an optimum solution is not certain in FCM. The initial cluster centres affect the performance of the algorithm and so to determine the initial cluster centre, another fast algorithm is used, or the same algorithm is run multiple times using different set of initial clusters.

5. Dataset Description

The datasets used in the analysis includes COCOMO dataset, Deshernais dataset, Maxwell dataset and IKH dataset. Whereas IBM, Kemerer, and Hallmark datasets are combined together as IKH dataset.

The COCOMO dataset is available from the historic projects of NASA. The developer determines the various rating levels depending on the attributes of the input, output and various other parameters. e.g., if the failure of the software causes slight inconvenience, then the developer may fix a very low rating to the parameter, and set the effort multiplier to be 0.82. If it is easy to recover from the software failure, then the developer may fix the rating level to be nominal and set the multiplier to one. In case, if the failure risks a human life, the developer may fix a high rating and set the effort multiplier say 1.26. Thus, the developer fix different effort multipliers based on the consequences of failures or some other factors. A more detailed description of fixing their scale factors and effort multipliers is shown in [4, 25].

The Deshernais dataset includes eight input parameters like team experience, manager experience etc., and an output effort for each project. It totally consists of 77 projects of which 62 projects are used for training and 15 projects are used for testing. The above dataset is available in the promise data repository.

Kathrina Maxwell generated the Maxwell dataset and was the most recent among all the datasets. It makes use of the categorical features and consists of 62 projects, among which 44 projects are used for training and 18 projects are used for testing. The dataset is available in the promise data repository.

Function points and lines of code makes up the IKH dataset [11]. It consists of 24 IBM projects, 15 Kemerer projects and 28 Hallmark projects. Where 17 of IBM, 11 of Kemerer and 20 of Hallmark projects are used for training, and 7 of IBM, 4 of Kemerer and 8 of Hallmark projects are used for testing. Table 1, Table 2 and Table 3 shows the entire datasets of IBM, Kemerer and Hallmark dataset.

Table 1. IBM dataset.

Sl.no	Actual Effort-hours (1000)	Function Points	Lines of code
1	43.62	1217.1	253.6
2	12.54	507.3	40.5
3	168.31	2306.8	450
4	13.21	788.5	214.4
5	51.12	1337.6	449.9
6	12.77	421.3	50
7	3.53	99.9	43
8	19.81	933	200
9	17.63	1592.9	289
10	10.94	240	39
11	39.32	1611	254.2
12	35.07	789	128.6
13	23.86	690	161.4
14	37.53	1347.5	164.8
15	10.62	1044.3	60.2

Table 2. Kemerer dataset**.**

Sl.no	Actual Effort hours (1000)	Function Points	Lines of code
1	102.4	1750	130
2	105.2	1902	318
3	11.1	428	20
4	21.1	759	54
5	28.8	431	62
6	10	283	28
7	8	205	35
8	4.9	289	30
9	12.9	680	48
10	19	794	93
11	10.8	512	57
12	2.9	224	22
13	7.5	417	24
14	12	682	42
15	4.1	209	40
16	15.8	512	96
17	18.3	606	40
18	8.9	400	52
19	38.1	1235	94
20	61.2	1572	110
21	3.6	500	15
22	11.8	694	24
23	0.5	199	3
24	6.1	260	29

Table 3. Hallmark dataset

Sl.no	Actual Effort-hours (1000)	Function Points	Lines of code
1	0.59	73	5
2	0.7	121	11.93
3	0.86	111	9.83
4	0.88	55	6.93
5	0.92	94	12
6	0.96	59	9.38
7	1.18	72	26.82
8	1.18	144	17.74
9	1.19	67	12.71
10	1.32	87	35.28
11	1.55	320	25.64
12	1.75	86	10.3
13	1.91	77	22.98
14	2.18	108	35.5
15	2.26	148	26.93
16	2.26	174	44
17	2.37	341	17.09
18	2.44	684	19.25
19	3.95	697	70.39
20	4.02	507	106
21	4.18	170	35.47
22	4.66	314	49.52
23	5.01	293	66
24	6.84	434	64.18
25	7.57	738	28
26	11.42	1206	50.38
27	13.14	791	72.75
28	23.3	1284	126.33

6. Experimentation

In a multilayer feed forward network each node performs a particular function on its input data and is said to be an adaptive in nature. The node function of such a network varies from one node to another. The links that connect the adaptive network are not associated with the weights.

The experiment uses four different types of datasets. The datasets are divided into two as training and testing datasets. The dataset are divided such that 70% is used for training and 30% is used for testing. The attributes of the project is given as the input to the layer 1 of the ANFIS model and the effort is obtained as the output of the network. ANFIS learns the relation between the input and output with the same knowledge it used for testing.

The simulation is carried out in MATLAB 10b environment. In an ANFIS network, the weights may take up any arbitrary values, so that there is an opportunity for obtaining assorted resolutions. To avert this problem, the entire network is simulated for 50 iterations and averaged their errors.

The subtractive clustering is created using 'genfis2' and the radii is fixed as 0.9. The number of epochs is set as 10, decreased step size as 0.9 and increased step size as 1.1.

Similarly, in Fuzzy C means clustering for creating a fismat, a sugeno type fuzzy Inference system is used and the number of clusters is fixed as 35.

7. Evaluation Criteria

Two types of tests were performed to evaluate the performance of different effort estimation models. They are error test and statistical test.

7.1. Error Test

The error test has a set of evaluation parameters namely Mean Magnitude of Relative Error (MMRE), Magnitude of Relative Error (MRE), Probability of relative error less than 0.25 (PRED(25)) and Root mean square error (RMSE) of which Mean Magnitude of Relative Error (MMRE) is one of the most widely accepted evaluation criteria.

The Magnitude of Relative Error (MRE) is defined as follows:

$$MRE_i = \frac{|actualeffort_i - predictedeffort_i|}{actualeffort_i} \tag{17}$$

For each observation, the MRE value is calculated for which the effort is predicted. The MRE aggregation over multiple observations (N) can be obtained from the MMRE as follows:

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \tag{18}$$

$$PRED(25) = \frac{MRE \le 0.25}{N} \tag{19}$$

Consider the neural network output Y and the desired target T. Then Root mean square error (RMSE) is calculated using the equation:

$$RMSE = \sqrt{(Y-T)^2} \tag{20}$$

7.2. Statistical test

The statistical test has a set of evaluation parameters namely ERROR, Mean (μ) and standard deviation (σ) of error, skewness (Y_1) and kutosis (Y_2), The parameter error is given by:

$$ERROR = (Y - T) \tag{21}$$

Mean of the error is given by:

$$\mu = \frac{\sum Error}{N} \tag{22}$$

and the standard deviation is given by:

$$\text{Standard deviation } \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{23}$$

Skewness Y_1 is the ratio of the third central moment μ_3 to the cube of its standard deviation 'σ' given by:

$$Y_1 = \frac{\mu_3}{\sigma^3} \tag{24}$$

The ratio of the fourth central moment μ_4 to the fourth power of its standard deviation σ denotes the kurtosis of the random variable. Kurtosis Y_2 is given by

$$Y_2 = \frac{\mu_4}{\sigma^4} \tag{25}$$

8. Results and Discussions

The accuracy of the ANFIS based effort estimator is analysed using two types of tests namely the error test and the statistical test. In the Error test MMRE, RMSE, and PRED parameters are calculated, and in statistical test mean, standard deviation, skewness and kurtosis are calculated, for each of the methods under consideration.

The model with the lower MMRE and smaller standard deviation will be considered the best method. The mean should be such that it is closer to zero in order for the effort to be accurate.

Table 4 gives a comparative result of the evaluation parameters obtained for Elman Neural Network, Fuzzy C Means clustering based ANFIS and Subtractive clustering based ANFIS using Deshernais dataset. In the analysis, 15 of the Deshernais projects are used for testing. MMRE is the considered as the decisive factor, to determine the accuracy of the effort estimator. The testing results confirm that FCM has lower MMRE than Subtractive clustering and Neural Network. Thus for Deshernais dataset, FCM is the best predictor model compared to other models. Figure 3 and 4 shows the variation of the actual and estimated effort for the FCM and subtractive clustering based ANFIS using Deshernais dataset.

Table 4. Comparative results of FCM and subtractive clustering for Deshernais dataset.

	ELMAN Neural Network		**FCM**		**SC**	
Deshernais	**Training**	**Testing**	**Training**	**Testing**	**Training**	**Testing**
MMRE	0.4288	0.5721	1.447	0.412	3.369	0.732
RMSE	3163	5725	12.750	38.312	8.454	24.025
PRED	43.5484	13.3333	45.161	40	35.483	13.333
Mean	678.3116	3463	-1.488	-8.069	1.985	11.335
Std.Dev	3115	4719	12.766	38.767	8.284	21.926
skewness	-0.5328	-0.3474	-4.222	-2.131	2.894	1.1446
Kurtosis	4.9895	2.1898	28.080	6.605	15.499	2.965

Table 5. Comparative results of FCM and subtractive clustering for Nasa dataset.

	ELMAN Neural Network		**FCM**		**SC**	
NASA Dataset	**Training**	**Testing**	**Training**	**Testing**	**Training**	**Testing**
MMRE	0.2171	1.1528	0.8768	1.623	1.17	1.84
RMSE	466.7348	1594	25.111	12.49	78.4	98.35
PRED	74.1935	15.873	29.166	26.315	25.45	17.47
Mean	-16.8969	-333.842	5.8725	5.0627	9.07	7.28
Std.Dev	468.9569	1571	24.673	11.73	36.78	22.83
skewness	-5.714	-4.8659	3.1497	0.2511	5.231	0.934
Kurtosis	53.5605	29.4835	12.765	2.6088	18.45	4.56

Table 6. Comparative results of FCM and subtractive clustering for IBM, Kemerer, Hallmark dataset.

	ELMAN Neural Network		FCM		SC	
IBM, Kemerer and Hallmark	Training	Testing	Training	Testing	Training	Testing
MMRE	0.4536	0.7031	1.447	0.412	3.369	0.732
RMSE	7.5631	11	421.41	1370.9	481.60	1362.7
PRED	43.75	42.1053	62.365	17.460	63.440	15.873
Mean	0.3173	-1.066	-68.43	-216.8	-89.88	-235.6
Std.Dev	7.6364	11	418.08	1364.5	475.71	1352.9
skewness	-1.8424	-0.3064	-3.088	-5.060	-6.065	-5.140
Kurtosis	11.4862	4.629	18.417	29.679	48.507	30.217

Table 7. Comparative results of FCM and subtractive clustering for Maxwell dataset.

	ELMAN Neural Network		FCM		SC	
Maxwell	Training	Testing	Training	Testing	Training	Testing
MMRE	0.396	1.3748	0.6472	0.3706	0.008	0.5021
RMSE	5796	6537	4043.6	6117.6	38.25	8163
PRED	48	5.5556	38.636	38.888	100	27.777
Mean	586	-2350	13.802	-1143	1.4	1738.9
Std.Dev	5833	6278	4090.3	6184.1	38.67	8206.8
skewness	-3.3341	-1.2776	0.0959	0.2344	0.3112	1.7001
Kurtosis	20.3954	5.5138	4.9149	3.7047	20.742	5.3684

Table 8. Comparative results of FCM and subtractive clustering for Kitchenham dataset.

	ELMAN Neural Network		FCM		SC	
Kitchenham	Training	Testing	Training	Testing	Training	Testing
MMRE	0.4132	0.5595	2.7851	0.5407	0.6088	0.9477
RMSE	1474	12721	2526	1722	1755	1735
PRED	42	22.2222	35	15.5556	38	8.8889
Mean	-3	-184	-457.781	52.7423	-221.045	837.9259
Std.Dev	1482	12864	2497	1753	1750	1753
skewness	-1.3132	-6.3065	-3.3529	-6.3868	-3.1461	-6.3923
Kurtosis	11.781	41.5538	17.0358	42.2231	16.7363	42.2534

Table 9. Comparative results of FCM and subtractive clustering for Telecom dataset.

	ELMAN Neural Network		FCM		SC	
Telecom	Training	Testing	Training	Testing	Training	Testing
MMRE	0.5222	0.5516	0.2942	0.5053	0.368	0.7337
RMSE	258	40	119	6.97	187	7.35
PRED	53.85	50	38.4615	75	46.1538	75
Mean	-97	-18	-3.7863	0.4263	46.5737	2.2266
Std.Dev	249	41	123	8.03	189	8.08
skewness	-2.1271	-1.1096	-0.7606	0.2519	0.4173	-0.0356
Kurtosis	6.6164	2.3019	3.2374	2.0041	4.6607	1.9484

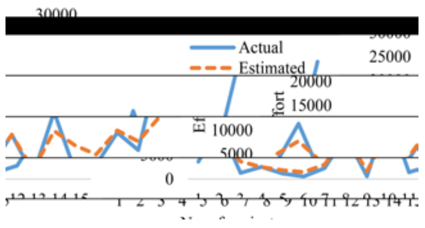

Figure 3. Effort estimation by FCM based ANFIS and data from Deshernais dataset.

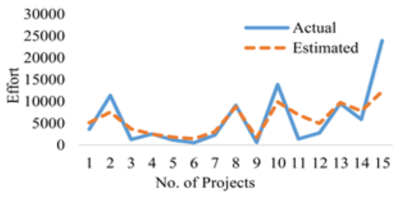

Figure 4.Effort estimation by Subtractive clustering based ANFIS and data from Deshernais dataset.

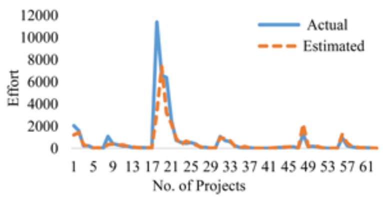

Figure 5. Effort estimation by FCM based ANFIS and data from NASA dataset.

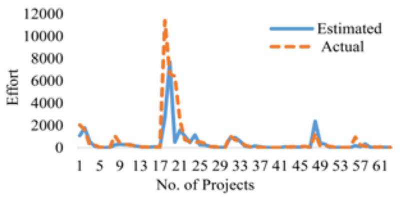

Figure 6. Effort estimation by subtractive clustering based ANFIS and data from NASA dataset.

Table 5 gives a comparative result of the evaluation parameters obtained for Elman neural network, Fuzzy C Means clustering based ANFIS and subtractive clustering based ANFIS using NASA dataset. In the analysis, 19 projects are used for testing. MMRE is the considered as the decisive factor, to determine the accuracy of the effort estimator. The results show a huge difference between the MMRE obtained for FCM and other methods. The MMRE obtained for FCM based ANFIS is 0.412, whereas SC and Elman Neural network show higher values. Figure 5 and 6 shows the variation of the actual and estimated effort for FCM and subtractive clustering based ANFIS. Table 6 gives a comparative result of the evaluation parameters obtained for Elman neural network, Fuzzy C Means clustering based ANFIS and subtractive clustering based ANFIS using the combined IBM, Kemerer and Hallmark dataset. MMRE is the considered as the decisive factor, to determine the accuracy of the effort estimator. The testing results show that MMRE for FCM base ANFIS is 1.623 compared to 1.687 of Elman neural network and 1.84 of subtractive clustering based ANFIS. Figure 7 and 8 shows the variation of the actual and estimated effort for FCM and subtractive clustering based ANFIS using IBM, Kemerer and Hallmark dataset.

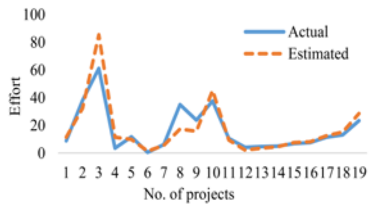

Figure 7. Effort Estimation by FCM based ANFIS and data from IBM, Kemerer and Hallmark dataset

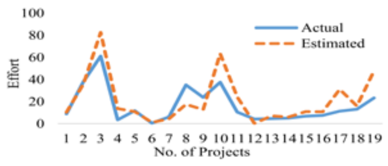

Figure 8. Effort Estimation by Subtractive clustering based ANFIS and data from IBM, Kemerer, and Hallmark dataset.

Figure 9 gives the variation of estimated effort and Actual Effort. Table 7 gives a comparative result of the evaluation parameters obtained for Elman Neural network, Fuzzy C Means clustering based ANFIS and Subtractive clustering based ANFIS using Maxwell dataset. The testing results show that the MMRE obtained for FCM based ANFIS is 0.37, which if far better than the MMRE values produced by subtractive clustering based ANFIS and Elman neural network given by 0.50 and 1.01 respectively. Figure 9 and 10 shows the variation of actual and estimated effort for FCM and subtractive clustering based ANFIS using Maxwell Dataset.

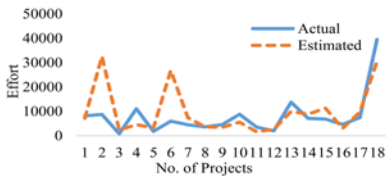

Figure 9. Effort Estimation by FCM based ANFIS and data from Maxwell dataset.

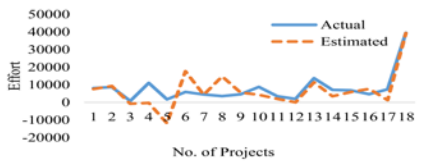

Figure 10. Effort Estimation by subtractive clustering based ANFIS and data from Maxwell dataset

Table 8 gives a comparative result of the evaluation parameters obtained for Elman neural network, Fuzzy C Means clustering based ANFIS and Subtractive clustering based ANFIS using Kitchenham dataset. The testing results show that the MMRE obtained for FCM based ANFIS is 0.54, which if far better than the MMRE values produced by subtractive clustering based ANFIS and Elman neural network given by 0.94 and 0.55 respectively. Figure 11and12 shows the variation of actual and estimated effort for FCM and subtractive clustering based ANFIS using Kitchenham Dataset.

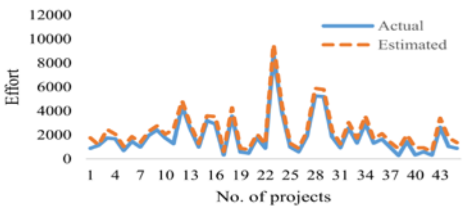

Figure 11. Effort Estimation by FCM based ANFIS and data from Kitchenham dataset.

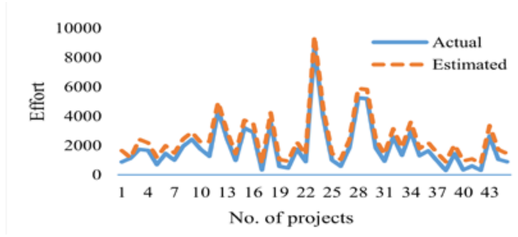

Figure 12. Effort Estimation by Subtractive Clustering based ANFIS and data from Kitchenham dataset.

Table 9 gives a comparative result of the evaluation parameters obtained for ELMAN Neural network, Fuzzy C Means clustering based ANFIS and Subtractive clustering based ANFIS using Telecom dataset. The testing results show that the MMRE obtained for FCM based ANFIS is 0.50, which is better than the MMRE values produced by subtractive clustering based ANFIS and neural network given by 0.73 and 0.55 respectively. Figure 13and 14 shows the variation of actual and estimated effort for FCM and subtractive clustering based ANFIS using Telecom Dataset.

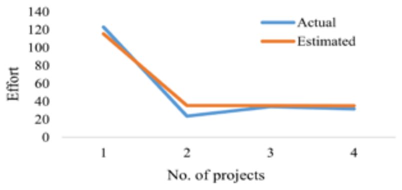

Figure 13. Effort Estimation by FCM based ANFIS and data from Telecom dataset.

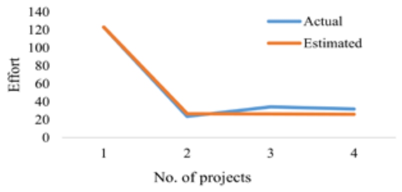

Figure 14. Effort Estimation by Subtractive Clustering based ANFIS and data from Telecom dataset.

9. Conclusion

Adaptive Neuro Fuzzy Inference System used here brings together the benefit of both the neural network and fuzzy logic. It is found that the Neuro fuzzy method facilitates better functionality than the standalone neural network and fuzzy logic systems. FCM and subtractive clustering based ANFIS is used in the effort estimation process. The ANFIS based effort estimation is compared with the Elman network based effort estimation. It is found that the ANFIS based estimation works well. The MMRE of ANFIS based estimation is less compared to the Elman network. Among the two ANFIS based estimation FCM works well than subtractive clustering. This is because the MMRE of FCM based ANFIS is lower than the subtractive clustering based ANFIS. Thus, ANFIS based on FCM has a better functionality over Elman network and subtractive clustering.

References

[1] Atterzadeh .I and Hock ow. S, "A novel algorithmic cost estimation models based on soft computing Technique". *Journal of Computing,* vol. 6, pp. 117-125, 2010.

[2] Azzeh. M, Neagu. D, Cowling. P. I, "Analogy-based software effort estimation using Fuzzy numbers,*" The Journal of Systems and Software,* vol. 84, pp. 270-284, 2011.

[3] Baxter. K, "Understanding software project estimates," *The Crosstalk*, pp. 26 – 29, 2009.

[4] Boehm. B, "COCOMO II: Model Definition Manuel. Version 2.1," *Center for Software Engineering,* USC, 2000.

[5] Chikako, Koten. V, "Bayesian statistical models for predicting software development effort,*" The Information Science Discussion Paper Series,* 2005.

[6] Chiu. S, "*Method and software for extracting Fuzzy classification rules by subtractive clustering*" Fuzzy Information Proceeding Society, Biennial Conference of the North American, pp. 461-465, 1996.

[7] Chulani. S, Boehm. B, Steece. B, "Bayesian analysis for empirical software engineering cost models," *IEEE transaction on software Engineering,* vol. 25, pp. 573-583, 1999.

[8] Cuauhtemoc Lopez-Martin, Claudia Isaza, Arturo Chavoya, "Software development effort prediction of industrial projects applying a general regression neural network", Emperical software engineering, Vol.17, PP.738-756, 2012.

[9] Geetha. N, Moin. U, Arvinder. K, "Grey relational effort analysis technique using regression methods for software estimation", *The International Arab Journal of Information Technology,* vol. 11, no. 5, Sep. 2014.

[10] Gray. A. R, Macdonnel. S. G, "A comparision of alternative to regresion analysis as model building technique to develop predictive equation for software metrics," *Information science discussion paper series,* 1996.

[11] Heiat. A, "Comparison of artificial neural network and regression models for estimating software development effort," *Information and software technology,* vol. 44, pp. 911-922, 2002.

[12] Huang. X, Ho. D, Ren. J, Carpertz. L. F, "Improving a COCOMO model using a neuro fuzzy approach," *Applied soft computing,* vol. 7, pp. 29 -40, 2007.

[13] Idri. A, zakarani. A, zahi. A, "Design of radial basis function neural networks for software effort estimation", *International journal of computer science Issues*, Vol.7, No.4, PP. 11-17, July 2010

[14] Jang. J. S. R, "ANFIS: Adaptive network based fuzzy Inference system,*" IEEE transaction on System. Man and cybernetics,* vol. 23, pp. 665-685, 1993.

[15] Jang. J. S. R, Sun. C, Mizutani. E. *Neuro – Fuzzy and Soft Computing.* Prentice-Hall, 2006;

[16] Jorgenson. M and shepperd. M, "A systematic review of software development cost estimation studies," *IEEE transaction on software engineering,* vol. 33, pp. 33-53, 2007.

[17] Kalichanin-Balich. I, Lopez-Martin. C, "*Applying a feed forward neural network for predicting software development effort of short-scale projects,*" Eighth ACIS International Conference on Software Engineering Research Management and Applications, pp. 269-275, 2008.

[18] Li. Y. F, Xie. M, Goh. T. N, "Adaptive ridge regression system for software cost estimating on multi collinear dataset," *The Journal of System and software,* vol. 83, pp. 2332-2343, 2010.

[19] Little. T, "Schedule estimation and uncertainty surrounding the cone of uncertainty," *IEEE software*, vol. 23, pp. 48-54, 2006.

[20] Marza. V, Seyyadi. A, and Capretz. L. F, "Estimating development time of software projects using a neuro fuzzy approace," *World academy of science Engineering and Technology,* vol. 46, pp. 575-579, 2008.

[21] Murray. J. P, "Managing IT project development hurdles. systems development management," 2001.

[22] Ochodek. M, Nowrocki. J, Kwarciak. K, "Simplifying effort estimation based on use case points," *Information and software technology,* vol. 53, pp. 200-213, 2011.

[23] Palival. M, Kumar. U. A, "Neural networks and statistical techniques - a review of applications," *Expert system with application,* pp. 2-17, 2009.

[24] Prasad Reddy P.V.G.D, Sudha.K.R, Rama Sree.P and Ramesh S.N.S.V.S.C, Software effort estimation using radial basis and generalized regression neural networks," Jourrnal of Computing, vol.2, No.5, pp.87-92, 2010.

[25] Praynlin. E, Latha. P, "Minimal Resource Allocation Network (MRAN) based software effort estimation," *International review on computer and software,* vol. 8, No. 9, pp. 2068-2074, 2013.
[26] Sadiq. M, Asim. M, Ahmed. J, Kumar. V, Khan. S, "Prediction of software cost estimation: A case study". *International journal of modelling and optimization*, vol. 1, pp. 37-43, 2011.
[27] Shukla. K. K, "Neuro genetic prediction of software development effort," *Information and software technology,* vol. 42, pp. 701-713, 2000.
[28] Walkerden. F, Jeffery. R, "An empirical study of analogy-based software effort estimation," *Empirical Software Engineering,* vol. 4, pp. 135-158, 1999.
[29] Witting. G, and Finnie. G, "Estimating software development effort with connectionist models," *Information Software Technology,* vol. 39, pp. 369-476, 1997.
[30] Witting. G, and Finnie. G, "Using artificial neural networks and function points to estimate 4GL software development effort," *Journal of Information Systems*, vol. 1, pp. 87-94, 1994.
[31] Yan Fu Li, Min Xie, Thong – Ngee Goh, "Adaptive ridge regression system for software cost estimating on multi-collinear datasets", The Journal of System and software, vol.83, pp.2332-2343, 2010

Praynlin: Research scholar in Department of Computer science and Engineering, Government college of Engineering, Tirunelveli. He has received his master's degree in Applied Electronics from Noorul Islam University. He graduated from Anna university in Electronics and communication Engineering. His area of interest are software cost estimation and neural networks.

Latha Associate Professor in Government college of Engineering, Tirunelveli. She has received her master's degree in computer science and Engineering from Bharathiar university. She graduated from Madurai Kamaraj university in Electrical and Electronics Engineering. She has published her research work in 4 International Journals, 6 National level conferences and more than 40 national level conferences. Her field of specialization is image processing.