EMPIRICAL APPLICATION OF SIMULATED ANNEALING USING OBJECT-ORIENTED METRICS TO INCREASE THE ACCURACY IN SOFTWARE EFFORT ESTIMATION

Sheenu Rizvi[1], Dr. S.Q.Abbas[2] and Dr.Rizwan Beg[3]

[1]Department of Computer Science,Amity University, Lucknow,India,
[2]A.I.M.T., Lucknow, India and
[3] R B Group of Institutions, Agra, India

ABSTRACT

The work is about using Simulated Annealing Algorithm for the effort estimation model parameter optimization which can lead to the reduction in the difference in actual and estimated effort used in model development.

The model has been tested using OOP's dataset, obtained from NASA for research purpose.The data set based model equation parameters have been found that consists of two independent variables, viz. Lines of Code (LOC) along with one more attribute as a dependent variable related to software development effort (DE). The results have been compared with the earlier work done by the author on Artificial Neural Network (ANN) and Adaptive Neuro Fuzzy Inference System (ANFIS) and it has been observed that the developed SA based model is more capable to provide better estimation of software development effort than ANN and ANFIS.

KEYWORDS

COCOMO, ANN, ANFIS, SA,KC1.

1.INTRODUCTION

Software Effort Estimation [1] is the approximate calculation about the work to be performed in order to develop a software project. It is measured in man months, meaning thereby the amount of work done by a man in one month. When salaries of the employees are factored into effort estimation, it leads to cost estimation. However effort is first estimated before cost so as to make the data used to be more compatible throughout the time period involved, location and industry domain. Like ten year old data can be used without adjusting for inflation. In software engineering effort is utilized to indicate measure of utilization of workforce and is characterized as aggregate time that takes individuals from an advancement group to perform a given assignment. It is normally communicated in units, for example, man-day, man-month, man-year. This characteristics is vital from the point of view that it forms the basis for determining the various features vital for software development. The most common reasons for estimation of effort are as follows:

DOI:10.5121/ijcsa.2015.5603

1. Project approval. Effort estimation is determined before starting any project so that they can be completed successfully.
2. Project management. Planning and management of the projects by the project managers require effort estimation for project completion.
3. Understanding by the development team members. Responsibility by each team member and understanding by the team of their activities helps in the efficient performance by the development team.
4. Effort estimation is used for defining project task. Effort estimation accuracy determination is still a challenge, even though it was launched by Brooks in his work "The Mythical Man Month" long back. Although lots of money and various activities has been invested for effort estimation but still there is a scope for improvement. Different authors classify effort estimation methods differently, which includes Empirical parametric (algorithmic) estimation models; Empirical non-parametric estimation models; Expert estimation; Analogue estimation models; etc.

2.ANN MODEL FOR EFFORT ESTIMATION

Models are required in almost all area of planning and management. They enable the response and behaviour of a system to be investigated under various physical conditions, which is generally either impossible or infeasible to do in the real world. Therefore, they play a vital role in areas such as Software cost Estimation and Effort Estimation. The types of models currently used to model systems can be broadly categorised into three main groups: physically-based, conceptual models and empirical model [3][2].

Physically-based models aim to represent the underlying physics of the system by using a series of partial differential equations to describe the change of state of the system over time. However, the development of a physically-based model is very data intensive, requiring spatial and temporal data to describe the physical characteristics of the system [3]. These types of data are generally not available; therefore, in many cases intensive data collection programs are required which can be both costly and time consuming. Secondly, there still remain many components of a system that are only poorly understood or are simply too complex to model accurately with mathematical relationship. Consequently, even the most complex physically-based models are not adequate to tackle the problems under investigation. Physical Based models limitations can be overcome by conceptual model, which deal with simplified description of the physical mechanism, representing only its important features. Conceptual models are therefore simpler than their physically-based counterparts, requiring fewer parameters, and are also less data intensive. However both conceptual and physical-based models require continuous data sets. If the data are missing, which is often the case, and then errors are introduced into the models.

2.1.Algorithm for ANN Model:

STEP 1:
For given inputs, variable output is calculated by the network, based on the initial random weights selection.

STEP 2:
Next calculate the error for neuron B. This error is the difference between the obtained error minus the observed error, i.e :-

$$\mathbf{ErrorB = OutputB\ (1\text{-}OutputB)(TargetB - OutputB)}$$

The “Output(1-Output)” is applied to cater to the sigmoid function being used. On the other hand for using only threshold neuron it would be (Target –Output).

STEP 3:
New trained weights, say W+ΔW are applied and initial weights are $W_{AB.}$

$$\mathbf{W+\Delta W = W_{AB} + (ErrorB \times OutputA)}$$

This weight change is that of Neuron A. In the same way other weights in the output are calculated.

STEP 4:
Next the error for the hidden layer neurons are calculated. For this back propagation technique is used. Here the error is back calculated from the outer layer neurons to the hidden layer neurons. Take for example the case where errors from B and C are back propagated to A, where error is given as below

$$\mathbf{Error\ A = Output\ A\ (1 - Output\ A)(ErrorB\ W_{AB} + ErrorC\ W_{AC})}$$

Where, "Output (1 - Output)" is due to sigmoid squashing function.

STEP 5:
Once the error for the hidden layer neurons has been calculated, repeat STEP 3 to change the hidden layer weights. By successive repetition one can train a network of any number of layers.

3.ANFIS MODEL DEVELOPMENT

An FIS uses the human knowledge by incorporating it in rule base and data base and thus performing fuzzy reasoning for drawing inference about the output value. The use of fuzzy if-then rules and membership function types can only be determined if one has prior knowledge about the system under consideration. However, there are still two basic but important problems concerning the preparation and manipulation of knowledge.

Firstly, lack of knowledge about converting the human expertise into knowledge base of FIS and secondly, in order to reduce the difference between the observed values and predicted values, there is need for learning algorithm for tuning the membership function [8][9]. These two problems greatly restrict the application domains of FIS. On the other hand, Neural Network modelling does not rely on human expertise. Instead, it employs a learning procedure and a given training data set to solve a set of parameters (i.e. weights) such that the required functional behaviour is achieved. No effective methods have been proposed to determine the initial weight values and network's configuration (e.g. number of hidden layers and hidden nodes).

Thus the drawbacks pertaining to these two approaches seems complimentary. Hence it is logical to use the combined fuzzy logic and ANN models. In other words, the integrated approach, or neuro-fuzzy modelling, should incorporate the three most important features,

1. Meaning and concise representation of structured knowledge.
2. Efficient learning capability to identify parameters.
3. Clear mapping between parameters and structured knowledge.

As seen earlier, fuzzy systems poses to the developers the following problems:

- Rules. Determination of fuzzy if-then rules is vital, which is time consuming, having many inherent problems because it is acquired from the expert knowledge.

- Membership functions. It is important because without it fuzzy set cannot be formed. It is therefore necessary to determine its type and parameters.

The ANFIS methodology takes in the tenets and enrolment capacities from information. ANFIS is a versatile system. A versatile system is system of hubs and directional connections. Connected with the system is a learning standard - for instance back engendering. It's called versatile on the grounds that some, or all, of the hubs have parameters which influence the yield of the hub. These systems are taking in a relationship in the middle of inputs and yields.
Versatile systems covers various distinctive methodologies however for our reasons we will research in some subtle element the technique proposed by Jang known as ANFIS.

The ANFIS building design is demonstrated as follows. The round hubs speak to hubs that are altered while the square hubs are hubs that have parameters to be learnt.

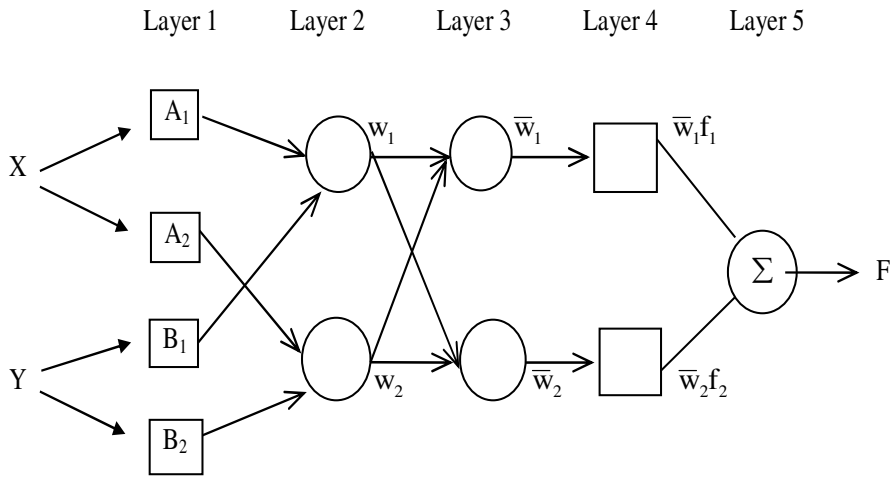

Fig 1. Two rule Sugeno System ANFIS architecture

The form of the Two Rule Sugeno ANFIS is as follows:

$$If\ x\ is\ A_1\ and\ y\ is\ B_1 \quad THEN\ f_1 = p_1x + q_1y + r_1$$
$$If\ x\ is\ A_2\ and\ y\ is\ B_2 \quad THEN\ f_2 = p_2x + q_2y + r_2$$

For system training, there is a forward pass and a backward pass. The forward pass proliferates the information vector through the system layer by layer. In the regressive pass, the mistake is sent back through the system in a comparative way to backpropagation.

Layer 1
The output of each node is:

$$O_{1,i} = \mu_{A_i}(x) \qquad for\ i = 1,2$$
$$O_{1,i} = \mu_{B_{i-2}}(y) \qquad for\ i = 3,4$$

So, the $O_{1,i}(x)$ is essentially the membership grade for x and y.

The membership functions could be anything but for illustration purposes we will use the bell shaped function given by:

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}}$$

where a_i, b_i, c_i are parameters to be learnt. These are the premise parameters.

Layer 2

Every node in this layer is fixed. This is where the t-norm is used to 'AND' the membership grades - for example the product:

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1,2$$

Layer 3

Layer 3 contains fixed nodes which calculates the ratio of the firing strengths of the rules:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}$$

Layer 4

In this layer nodes are adaptive and calculate the consequent of the rules:

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i)$$

The parameters in this layer (p_i, q_i, r_i) are to be determined and are referred to as the consequent parameters.

Layer 5

Its a single node that calculates the overall output:

$$O_{5,i} = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

In this way input vector is passed through the network layer by layer. And ANFIS is able to learn the premise and consequent parameters for the rules and membership function.

3.1.Hybrid Learning Algorithm

It combines Steepest Descent and Least Squares Estimation (LSE). Thus for a network, if the premise parameters are fixed, in the consequent parameters the output is linear.

The parameter sets are split into three parts

S = set of total parameters

S_1 = set of premise (nonlinear) parameters

S_2 = set of consequent (linear) parameters

Hence, ANFIS uses a two pass learning algorithm:

- **Forward Pass.** Here S_1 is unmodified and S_2 is computed using a LSE algorithm.
- **Backward Pass.** Here S_2 is unmodified and S_1 is computed using a gradient descent algorithm such as back propagation.

Thus hybrid learning algorithm uses steepest descent and least squares in combination so as to do parameter adaptation for the network.

3.2.Algorithm used:

The Forward Pass

Step1: First, Present the input vector
Step2 : Calculate the node outputs layer by layer
Step 3: Repeat for all data $\rightarrow A$ and Y formed
Step 4 S_2 parameter identification using Least Squares
Step 5: Computation of error for each training pair

Backward Pass

Step 1; Use of steepest descent algorithm for updating parameters in S_1 (back-propagation)
Step 2; S_2 parameter values found by using the present approach for given fixed values of S_1 the parameters are global optimum point for sure.

4.SIMULATED ANNEALING (SA) TECHNIQUE FOR MODELLING

It is a widely used evolutionary algorithms for combinatorial optimization. Evolutionary algorithm refers to any probabilistic algorithm whose concept is generated by evolutionary mechanisms found in biological species [5].SA is very well known heuristic algorithm. It uses the cooling and then freezing of a metal into low energy crystalline structure analogy [4]. During the optimization process, with probability using Metropolis principle, the solution is determined, during which the system temperature slowly decreases. When the annealing temperature is nearing zero, it is at this temperature that the solution is global best. [4].

Using SA for optimization problem solution, it is treated as an NLP problem, wherein the objective functions and constraints functions are expressed as independent variables.
The objective function is expressed as
Optimize f(x)

Such that ‘x’ exists within the n-dimensional feasible region D:
X ı D, where
D = {x | x >=0, $g_i(x)$ <=0, $_{hi}(x)$ = 0, i=1 to n}
Where f(x), $g_i(x)$ are real valued scalar functions,
vector x comprises the n principal variables for which the optimization is to be performed
f(x) = objective function, for which the optimal value of x leads to maximum value for f(x), and thus satisfying the given constraints.

The function f(x) is called to be objective function, for which the optimal value of x result in the maximum value for f(x), and these optimal values satisfy the given constraints.

4.1.Algorithm

Simulated Annealing
Begin
Initialize (T_0,N_0);
K: = 0;
Initial configuration S_i
Repeat procedure
Do L: =1 to Nk
Generate (S_j from S_i);
If $f(S_i) \leq f(S_j)$ do $S_i = S_j$
Otherwise
If $\exp\{(f(S_i)-f(S_j)/Tk) >$random [0,1] do $S_i = S_j$;
End do;
K = K+1;
Calculation of the length (Nk);
Determine control parameter (Tk)
Stopping criterion
End;
From the current state S_i with cost $f(S_i)$, a neighbor solution S_j, with cost $f(S_j)$ is generated by the transition mechanism. The following probability is calculated in performing the acceptance test:

$$\mathbf{P_T\{Accept\ S_j\} = 1} \quad \mathbf{if\ f(S_j) \leq f(S_i),\ or}$$
$$\mathbf{\exp\{(f(S_i)-f(S_j)/Tk)\},} \quad \mathbf{if\ f(S_j) > f(S_i)}$$

5.RESULT AND DISCUSSION

The overall objective of this work, as stated earlier in the introduction, was to use artificial intelligence techniques like Artificial Neural Network (ANN), Adaptive Neuro Fuzzy Inference System (ANFIS) for optimum model development for prediction of Development Effort Estimation using NASA project dataset, consisting of 63 software projects. Further, to develop an optimization model based on COCOMO model parameters using Simulated Annealing Technique. Next, to compare the results based on SA approach with ANN, ANFIS and COCOMO models proving the estimation utility of the SA approach.

Through the work presented, it was shown that models developed using both ANN and ANFIS techniques could be used to effectively address these issues. It is found that ANN and ANFIS models are very robust, characterised by fast computation, capable of handling the noisy and approximate data that are typical of the present data used. The results show that ANN models perform well even if the input data has noise and measurement errors. All these are scanned out during ANN training.

From the analysis of the results it was seen that the Effort Estimation prediction model developed using Feed Forward Neural Network technique with back propagation training algorithm has been able to perform well. For non-linear data, it is very good quantitative tool to forecast effort estimation. The work has been performed using MATLAB. Sixteen input variable have been used, of which fifteen are Effort Adjustment Factors and size of the project and Effort is the output variable. Although in all nine training algorithms were used for model development, but Levenberg-Marquardt training was able to perform better than the rest. Here it was noted that

increase in number of hidden neurons from 2 to 10 led to improvement in the model accuracy, but further increase led to the decline in it. Further, comparative study of COCOMO model and ANN model showed almost similar prediction accuracy as concluded.

ANFIS is the combination of ANN and Fuzzy logic and this makes it a powerful computational tool. This can handle both subjectivity and uncertainity in effort estimation in a scientific manner. The ANFIS model provides nonlinear modelling capabilities and requires no assumption of the underlying model. Now it is possible to express linguistic relationship between the output and input variables using fuzzy rules. Training the ANFIS models using ANN one can get missing fuzzy rules by extracting information from the existing data by doing extrapolation. Further, the adoption of Gaussian membership function for the development of ANFIS model show that it performs better than other MFs for optimum model development with the same number of epochs used.

Here the introductory parameters of the ANFIS are distinguished utilizing the subtractive bunching strategy. Gaussian enrolment capacities (given in before area) are utilized for each fuzzy set as a part of the fuzzy framework. The quantity of participation capacities and fuzzy principles required for a specific ANFIS is resolved through the subtractive grouping calculation. Parameters of the Gaussian enrolment capacity are ideally decided utilizing the cross breed learning calculation. Each ANFIS has been prepared for 10 epochs.

From the investigation of the outcomes, it was seen that the Effort Estimation forecast model created utilizing ANFIS method has possessed the capacity to perform well over COCOMO Model. This can be finished up from the investigation of the outcomes. The RMSE quality acquired from ANFIS model (112.638) is lower than those from COCOMO Model (532.2147). Further, it is seen that ANFIS model line intently takes after the genuine exertion line than those of COCOMO. This again depicted the superiority of ANFIS technique over COCOMO model for effort estimation. Next, from the analysis of the results on Simulated Annealing, it was seen that simulated annealing (SA) is a flexible technique for solving hard combinatorial problems. Its fundamental favourable circumstances over other nearby pursuit strategies are its adaptability and its capacity to approach worldwide optimality. It has a flexible calculation because of its independence from the prohibitive properties of the model. SA systems are effectively "tuned". For any sensibly troublesome nonlinear or stochastic framework, a given improvement calculation can be tuned to upgrade its execution and since it requires investment and push to get comfortable with a given code, the capacity to tune a given calculation for use in more than one issue ought to be viewed as a vital component of a calculation. It statistically guarantees finding an optimal solution. In the present work a new model based on the lines given by COCOMO using simulated annealing optimization technique has been developed, wherein the constant values, ‘a’ and ‘b’ are optimized so as to lead to a better solution method. An attempt is being made to make the observed value very near to predicted value of effort. This results in very low value of MRE. The dataset used has two input variables as Lines of Code (LOC) and Effort Adjustment Factor (EAF) and the output variable as Development Effort (DE).

A number of SA models, using different combinations of annealing function and parameter update functions have been developed for each of the three development mode, viz. semi-detached, embedded and organic mode. From the analysis of the results given under the heading “Results and Discussions”, it was seen that using combination of Fast annealing function algorithm and exponential parameter update function, the it was seen that equation for organic mode was found to be the best developed model, resulting in low RMSE value of 14.85 as compared to that of COCOMO model for the same development mode having RMSE value of 28.704. On the other hand, using Boltzmann annealing function and Boltzman parameter update function in combination, it was seen that Semi-Detached SA model (SD_BB) and Embedded

EM_BB was found to be the best optimized model, having an RMSE value of 81.58 and 226.214 as against that of COCOMO which is 402.7236 and 756.898 respectively. Further, using SA technique a single optimized model equation was developed for the whole set of data, as against the three equations developed earlier. Various combination of parameter update function and annealing functions were used for the development of SA model. The optimized model so developed has the best model equation corresponding to F_FB, with minimum values of RMSE and MRE. This led to the conclusion that SA model performed better than COCOMO model. Further consider the Tables 1 and Fig. 2 and 3 given below for comparative analysis of all the predictive models using ANN, ANFIS and SA TECHNIQUES.

Table 1. Comparative RMSE and MMRE values for all models

Model		RMSE	MMRE
COCOMO		532.2147	32.978
ANN		353.1977	83.0234
ANFIS	**Training**	0.00302	0.0000892
	Testing	2756.895	125.6756
	Complete	112.638	39.95
SA		*154.9344*	*26.286*
SA Single		177.9143	38.599

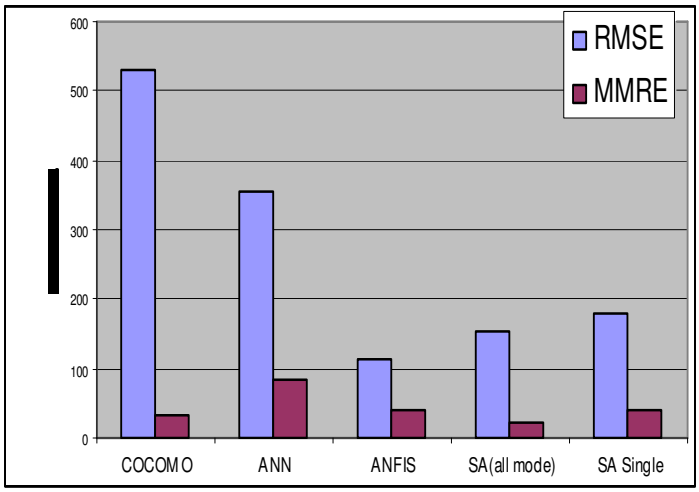

Fig. 2. Comparative plot of RMSE and MMRE val. for all models

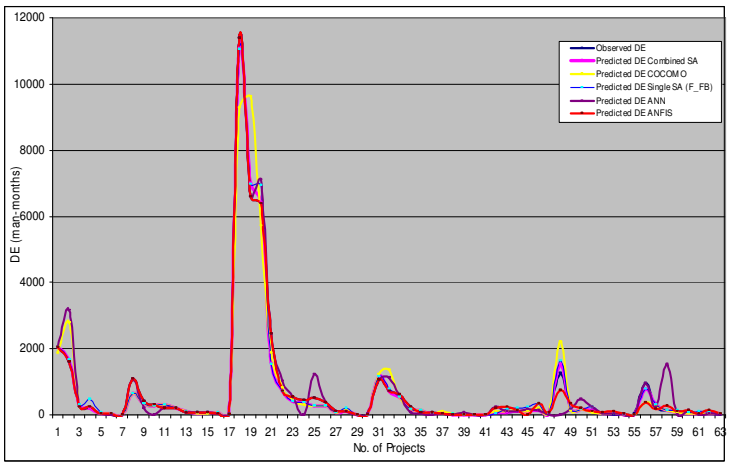

Fig. 3:Relative plot of Observed versus predicted DE for all models

From the perusal of the data given in the Table 1 above, it is clear that SA optimization technique has outperformed ANN, ANFIS and COCOMO models for better prediction model development. This has been clearly demonstrated in Fig. 2. The RMSE values for SA model is 154.9344 as compared to 353.1977 and 532.2947 for ANN and COCOMO respectively, although it is slightly more than that of ANFIS model. Hence overall it is superior to all other techniques. Also, MMRE values for SA model (26.286) is better than all other techniques, viz. ANN (83.0234), ANFIS (39.95) and COCOMO (32.978).

5.1.Effort Estimation Approach application on KC1 OOS database using Simulated Annealing:

The development of OOS effort estimation model using SA algorithm has been implemented trough programming in MATLAB using NASA project KC1 datasets. The modeling study is based on the SD Effort Estimation of the KC1 OOS. The parameter values of the estimation model are optimized in order to arrive at the optimum prediction model using Simulated Annealing Technique. The COCOMO Object Oriented metrics are in [6]. The major metric used are the Lines of Code (LOC) and HALSTEAD CONTENT Numeric (HCN). The details about the implementation and evaluation of Effort Estimation using Simulated Annealing have been dealt in detail in the upcoming sections.

5.1.1.Eight medium-sized systems developed by Basili et al. [10] were analysed and on its basis it was found that many Chidamber and Kemerer metrics were prone to fault.The relationship between OO metrics and the fault detection probability in system classes were empirically studied by Briand et al. It showed that perfect forecasting models could be developed to predict faulty classes.

In the present work, OO metrics given by Chidamber and Kemerer [11] have been empirically validated. Here Testing effort measured in terms of lines of code added or changed along with one more attribute found by linear regression analysis is the dependent variable. data set KC1 from the NASA Metrics Data Program [7] has been used. The data in KC1 was obtained for receiving/processing ground data and has been implemented in the C++. It consists of 145 classes comprising of 2107 methods, with a total of 40K lines of code. KC1 provides both class-level and method-level static metrics. Values of several metrics have been computed including seven

metrics given by Chidamber and Kemerer [11] at the class level. These OO metrics have been considered here for analysis.

We have collected data from web link http://promise.site.uottawa.ca/SERepository. The name of dataset is Class-level data for KC1 it was used for A. Gunes Koru and Hongfang Liu in the investigation of the Effect of Module Size on Defect Prediction Using Static Measures in 2005.Machine learning algorithms are used to predict the defective modules, cost and efforts on different NASA products CM1, JM1, KC1, KC2, and PC1 etc. This dataset consist of set of static measures which used in or work as predictor variables. It was seen that when measured using LOC, modules were found to be small.

5.1.2.We have observed that the KC1 database consist of a matrix data of 95x145 aray. It shows that there are 95 attributes evaluated by metric model suggested in [3] for 145 s/w projects. The 94 atttributes are numeric but the last one is defect it is non numeric i.e in terms of true or false. We have taken all the numeric attribute excluding the defect. There after the effort value is at 21st position in this database named as 'HALSTEAD_EFFORT '. This 145 values of effort is taken as output and the remaining 93 attributes are considered as prediction variables for determining effort. Since in our previous discussion we described effort estimation model, the SA based estimation relations were derived as a function dependent of line of codes (LOC) and EAF for COCOMO81 dataset. Hence for KC1 datasets we considered line of codes as one of the independent parameter and this attribute is located at 94 position in KC1 with name as ' sumLOC_TOTAL numeric'. [7] suggested that forecasted lines of code added or changed using SLOC metric are very much correlated with actual lines of code added or changed. Also, MARE and MRE values are minimum for SLOC metric. Hence, it is the best predictor of testing effort. Further, KC1 does not indicate any formulation for EAF measurement and our proposed mathematical model can estimate effort using two independent variables. One of the independent variable i.e. LOC is present in KC1 and we had to select any one of the attribute out of 92 attributes on which effort depends most. For this purpose we considered linear regression technique to determine the weightage coefficient in terms of effort dependency over the 92 attributes. Using linear regression we obtained the coefficient value of linear relation between efforts on other attributes. These coefficients are then sorted out in descending order to collect the most significant attributes which has major role in defining dependencies of effort. The top twenty sorted list of coefficients and respective attributes are given in table 2.

Table 2 :Regression coefficients for linear dependencies for effort in terms of OOS metric attributes in descending order.

Attribute name	Regression Coefficient
'avgHALSTEAD_CONTENT numeric'	1298119.17772377
' avgHALSTEAD_LEVEL numeric'	797070.001250424
' avgHALSTEAD_LENGTH numeric'	501048.430790699
'maxESSENTIAL_COMPLEXITY numeric'	53227.8294687621
' maxHALSTEAD_EFFORT numeric'	53227.7513680517
'sumESSENTIAL_COMPLEXITY numeric'	41403.4757097250
' sumHALSTEAD_EFFORT numeric'	41402.2043732386
' avgESSENTIAL_COMPLEXITY numeric'	21888.7110168278
' avgHALSTEAD_EFFORT numeric'	21888.6710406168
' minHALSTEAD_DIFFICULTY numeric'	75.7336314420953
' maxHALSTEAD_ERROR_EST numeric'	7.94134419196977
' maxLOC_EXECUTABLE numeric'	7.27342285875562
' sumHALSTEAD_LENGTH numeric'	4.66884388671975
' avgHALSTEAD_ERROR_EST numeric'	4.46541530501098

' avgLOC_EXECUTABLE numeric'	3.98409415852182
' sumDESIGN_COMPLEXITY numeric'	3.97549918810867
' sumHALSTEAD_PROG_TIME numeric'	3.90923046771693
' sumLOC_BLANK numeric'	2.40411231755162
' sumHALSTEAD_VOLUME numeric'	2.34191565431222
' sumHALSTEAD_ERROR_EST numeric'	2.28542559747626

It has been observed that the effort has highest dependency over' avgHALSTEAD_CONTENT numeric' because it has regression coefficient as 1298119 (aprrox.) . It has been concluded that 'avgHALSTEAD_CONTENT numeric' (HCN) can be our one of the attribute that can be considered for determining the effort.

After getting the independent variables as LOC and HCN we applied our SA based algorithm model for developing parametric model non linear equation of estimating effort as a function of LOC and HCN. The objective function for SA is given below:

Error =actual_effort - hcn*x1*loc^x2+x3 (1)

SA has derived x1 , x2 and x3 parameters for all the 145 softwares These parameters are retested for determining mean square error (m.s.e) when applied one by one for each s/w and final our algorithm has given an optimum value for x1 , x2 and x3 for estimating effort. The algorithm is applied several times at different annealing function and parameter update function and mse is evaluated at different lower bounds(Lb) and upper bounds(Ub) all the results are tabulated in table 3.

Table 3 :Parametric values for effort estimation model achieved by OOS metric based SA model equation

AF	**PUF**	**X1**	**X2**	**X3**	**mse**	**Lb**			**Ub**		
1	1	0.299999909639709	0.059999981373202	4.00000015797827	6.358	0.2	0.05	4	0.3	0.06	6
1	1	0.199999358079533	0.059999999748975 9	3.00000007250211	4.665	0.1	0.05	3	0.2	0.06	5
2	1	0.249999855987849	0.049999951446923 8	4.50000076337305	5.507	0.2	0.04	4.5	0.25	0.05	5.5
2	1	0.249999999916248	0.039999639277056 4	4.50000450793711	5.399	0.23	0.03	4.5	0.25	0.04	5
2	1	0.199999453250050	0.074999757288984 6	5.00000546760419	5.377	0.1	0.06	5	0.2	0.075	6
1	2	0.249999999843231	0.049999999848397 6	2.00000042958199	4.943	0.2	0.01	2	0.25	0.05	2.5
1	2	0.299999995810968	0.049999999643931 0	2.00000170539898	5.389	0.2	0.01	2	0.3	0.05	2.5
2	2	0.199999994144019	0.059999982712746 8	2.50000333817231	4.624	0.1	0.05	2.5	0.2	0.06	2.8
2	2	0.199999999989851	0.059999987779657 9	3.00000001400614	4.665	0.1	0.05	3	0.2	0.06	5
2	2	0.249999992722212	0.049999999799949 8	2.00000024249984	4.943	0.2	0.01	2	0.25	0.05	2.5

1	3	0.249999999583991	0.029999999293903	1.50000008027387	4.871	0.2	0.01	1.5	0.25	0.03	2
1	3	0.199999999629496	0.029999999920358	1.40000003086126	4.670	0.18	0.01	1.4	0.2	0.03	1.8
2	3	0.199999999728870	0.029999999644955	2.00000001107818	4.580	0.18	0.01	2	0.2	0.03	2.2
2	3	0.199999999999956	0.049999999999358	2.20000000967558	4.596	0.18	0.03	2.2	0.2	0.05	2.4
2	3	0.179999999444613	0.049999999777978476	2.20000000889153	4.488	0.16	0.03	2.2	0.18	0.05	2.4

From the table 3 it has been shown that by taking different combination of annealing function and parameter update function we can obtain different value of parameters x1 , x2 and x3 at which the effort can be estimated easily by using LOC and HCN metric attributes for OOS considered in KC1 database. Column 6 of table 3 represents the mse in estimated effort with respect to actual effort. In all the cases AF=1 indicates 'I' (Boltz) and 2= 'R' (fast) whereas PUF=1 is 'I'(Boltz) , 2 is 'R'(Fast) and 3 is 'E'(exponential). For all cases multiple times we applied our SA based methodology for finding x1,x2 and x3. It has been observed that least mse is obtained at AF=2 and PUF = 3 and it the mse is 4.48 at the value of x1=179999999444613,x2=0499999977978476 and x3=2.20000000889153 where lb is lower bound and ub is upper bound. In this way by using equation 1 we can get minimum error in estimation of effort at above mentioned values of x1,x2 and x3.

6.CONCLUSION

On comparing SA with ANN and ANFIS techniques for Software Development Effort estimation model development, it is seen that although Neuro-fuzzy method ANFIS presented in this thesis shows a good potential to model complex, nonlinear and multivariate problems. If however training not done properly the model will be poorly developed. Further it is also seen that ANFIS model is better than ANN model. Here, ANFIS has all the basic properties of ANN model but it simplifies the model building process. ANFIS model so developed also takes care of the trial and error procedure as used for developing the ANN model. Computational time is also reduced. By using subtractive clustering algorithm in ANFIS the initialization process is fast as compared to ANN where it required several stages of random selection. Further the number of training epochs required for convergence in ANFIS is far less than in ANN. The lower MMRE and RMSE obtained by the ANFIS method suggests its good generalization capability.

It is seen that for non linear optimization problems, like training of ANN network, the ANN technique leads to poor and inconsistent results. Thus instead of ANN technique which looks for local solution, SA technique has been used which searches for the global minimum. Hence it is concluded that poor ANN performance can be attributed to the limitations of back propagation during training procedure. This depends upon the initial weights selected and if not done properly then the training process might be trapped in local minima. Also, the angle plummet calculation is for the most part moderate in light of the fact that it requires little learning rates for stable learning. The force variety is normally speedier than straightforward slope plunge, on the grounds that it permits higher learning rates while looking after soundness, yet it is still too moderate for some down to earth applications. Proper selection of learning rate is a challenge. Too large value leads to unstable learning, whereas too small results in increased training time.

Further, superiority of SA over ANFIS can be judged by the fact that ANFIS operates under certain constraints.

These are that ANFIS is a great deal more mind boggling than the fuzzy interface frameworks, and is not accessible for the majority of the fuzzy interface framework alternatives. In particular, ANFIS just backings Sugeno-sort frameworks, and these must have the accompanying properties:

- Be first or zeroth request Sugeno-sort frameworks.
- Have a solitary yield, got utilizing weighted normal defuzzification. All yield participation capacities must be the same sort and either be direct or consistent.
- Have no tenet sharing. Diverse guidelines can't have the same yield participation capacity, to be specific the quantity of yield enrollment capacities must be equivalent to the quantity of principles.
- Have solidarity weight for every standard.

A blunder happens if your FIS structure does not consent to these requirements. After justifying above model performance we have developed effort estimation model for OOS using the metric attributes. For demonstration the capability of effort estimation we considered the KC1 database. This database consist of information of about 145 NASA application related OOS in terms of 95 attributes. We applied linear regression technique for determining the regression coefficients of respective attributes in evaluation of effort value by linear equation problem. It has been found that Halstead content numeric(HCN) attribute has highest regression coefficient. Thus it is concluded that effort has maximum dependency on HCN. Then a mathematical equation is generated for evaluating effort in terms of LOC and HCN for the database available in KC1.The performance is checked in terms of mse in actual effort and predicted effort at different combination of annealing function and parameter update function at different upper and lower bounds of parametric values in SA based approach. It is seen that lower MSE value is due to use of Annealing function and exponential PUF. The optimum parameter values are x1=0.179,x2=0.049 and x3=2.2, having MSE of 4.48. Hence it can be concluded that SA performed better than COCOMO and is also useful in case of OOS datasets.

REFERENCES

[1] Jovan Zivadinovic, et. al., (2011), “Methods of Effort Estimation in Software Engineering”, International Symposium Engineering Management And Competitiveness.

[2] Asce Task Committee On Application Of Artificial Neural Networks In Hydrology, (2000b), “Artificial neural networks in hydrology. II: Hydrologic applications”, Journal of Hydrologic Engineering, 5(2), pp 124–137.

[3] Asce Task Committee On Application Of Artificial Neural Networks In Hydrology, (2000a), “Artificialmm neural networks in hydrology. I: Preliminary concepts”, Journal of Hydrologic Engineering , 5(2), pp 115–123.

[4] P. J. M. Laarhoven and E. Aarts, Simulated Annealing: Theory and Applications, Reidel, Ordrecht, 1987.

[5] D. Henderson et al., "The Theory And Practice Of Simulated Annealing”, 287-319.

[6] Tang, M., Kao, M., and Chen, M.,(1999) “An Empirical Study on Object-Oriented Metrics”, IEEE Transactions on Software Engineering, 0-7695-0403-5.

[7] T.Gyimothy , R.Ferenc , I.Siket , “Empirical validation of object-oriented metrics on open source software for fault prediction”, IEEE Trans. Software Engineering, vol. 31, Issue 10, pp.897 – 910, Oct. 2005.

[8] Chen, D.W. and Zhang, J.P., (2005), “Time series prediction based on ensemble ANFIS”, Proceedings of the fourth International Conference on Machine Learning and Cybernatics, IEEE, pp 3552-3556.

[9] Jeffery R, Ruhe M,Wieczorek I, "Using Public Domain Metrics to Estimate Software Development Effort," In Proceedings of the 7th International Symposium on Software Metrics, IEEE Computer Society, Washington, DC, pp 16–27, 2001

[10] V.Basili, L.Briand, W.Melo, "A Validation of Object-Oriented Design Metrics as Quality Indicators", IEEE Transactions on Software Engineering, vol. 22 no.10, pp. 751-761, 1996.

[11] S.Chidamber and C.F.Kemerer, "A metrics Suite for Object-Oriented Design", IEEE Trans. Software Engineering, vol. SE-20, no.6, 476-493, 1994.

Authors

Sheenu Rizvi, is a Research Scholar in Integral University, U.P, India. Currently he is working as Assistant Professor in Amity School of Engineering and Technology, Amity University, Lucknow, India. His area of interest includes Software Engineering and Soft Computing.

Dr Syed Qamar Abbas got His M.S.(Computer Science) from BITS PILANI and PhD in Computer-Oriented study of Queueing models. Currently he is working as Director General of Ambalika Institute of Management and Technology, Lucknow, India. He has more than 26 years of teaching and research experience in the field of Computer Science and Information Technology. He has published more than 130 Research papers in International Journals & Conferences. 11 Ph.D have been awarded under his guidance.

Dr.Rizwan Beg got his Ph.D in Computer Science. & Engineering. from Integral University, Lucknow, U.P, India. Currently he is woking as Director, R B Group of Institutions, Agra. He has more than 17 years of experience which includes around 15 years of teaching experience. His area of expertise is Software Engineering., Requirement Engineering, Software Quality, and Software Project Management. He has published more than 80 Research papers in International Journals & Conferences. 02 Ph.D have been awarded under his guidance.