# Empirical Assessment and Optimization of Software Cost Estimation Using Soft Computing Techniques

**Gaurav Kumar and Pradeep Kumar Bhatia**

**Abstract** Software Engineering especially project planning, scheduling, monitoring and control are based on accurate estimate of the cost and effort. In the initial stage of Software Development Life Cycle (SDLC), it is hard to accurately measure software effort that may lead to possibility of project failure. Here, an empirical comparison of existing software cost estimation models based on the techniques used in those models has been elaborated using statistical criteria. On the basis of findings of empirical evaluation of existing models, a Neuro-Fuzzy Software Cost Estimation model has been proposed to hold best practices found in other models and to optimize software cost estimation. Proposed model gives good result as compared to other considered software cost estimation methods for the defined parameters in overall but it is also dependent on type of project, data and technique used in implementation.

**Keywords** Back propagation neural network (BPNN) · Constructive cost model (COCOMO) · Function point (FP) · Fuzzy logic (FL) · Genetic algorithm (GA) · Particle swarm optimization (PSO) · Software cost estimation (SCE)

## 1 Introduction

Software Cost Estimation (SCE) of a software project starts from initial phase of software development which includes generating proposal requests, analysis, contract negotiations, planning, scheduling, designing, implementation, maintenance, monitoring and control. The estimation process includes size and effort estimation, initial project scheduling and finally estimation of overall cost of the project.

G. Kumar (✉) · P.K. Bhatia
Department of Computer Science and Engineering, Guru Jambheshwar University of Science & Technology, Hisar, Haryana, India
e-mail: er.gkgupta@gmail.com

P.K. Bhatia
e-mail: pkbhatia.gju@gmail.com



R.K. Choudhary et al. (eds.), *Advanced Computing and Communication Technologies*,
Advances in Intelligent Systems and Computing 452,
DOI 10.1007/978-981-10-1023-1_12

**Table 1** Features and limitations of techniques used in SCE

| Technique used in SCE models | Features | Limitation |
|---|---|---|
| Analogy-Historical | Based on actual experience | Much information of historical projects is required |
| Expert judgment | Fast prediction, Easy to use | Dependency on experts |
| COCOMO | Common method | A large amount of data is required |
| ANN | Consistent, Ability of generalization | Training data dependency |
| FL | Flexible, Training not required | Hard to use |
| GA | Optimization | Initial values required |
| PSO | Optimization | Training data dependency |

Accurate estimation of software cost is necessary to complete project within time and budget and to prevent failure of software project. If effort estimation is done too low, it may lead to problems in managing project, delay in delivery, overrun of budget and low software quality. If effort estimation is done too high, it may cause business loss and inefficient use of resources. Accuracy is important while software estimation for developers as well as customers as it determines what, where, and when resources will be used, analyzes impact of requirement change etc. Various SCE models have been developed to manage software project's budget and schedule. Estimation models developed so far has their own significance or importance and is applicable for specific type of projects. So the criteria to evaluate accuracy of software estimation model are much important to successfully complete a software project. Techniques used in estimating software cost have their own features as well as limitations. Some of which have been described in Table 1.

## 2 Review of SCE Models Based on Used Technique

A lot of research has been carried out for implementation of SCE models using various methodologies. A brief overview of research work done in the past for developing SCE model has been discussed here.

While developing SCE model, ANN acts as a proven practical way that reduces the model's input space (and thus computational complexity and human effort) while maintaining the same levels of effort prediction accuracy. An automated SCE applied on COCOMO data set using Feed forward BPNN tested on COCOMO NASA 2 dataset may help project manager for fast and realistic estimation of software cost for project effort and development time [1, 2]. Matlab Neural Network tool box with data from multiple projects can be used to validate, train and simulate the network with observations that neural network performs

better than COCOMO; and Cascade correlation performs better than Neural Network [3]. BPNN model with COCOMO data works well for Small projects while neural network with Resilient Back Propagation is good for big projects [4]. Radial Basis Function Neural Network with K-means clustering algorithm can perform better in terms of accurate cost estimation [5]. A Neuro-Fuzzy Constructive Cost Model (COCOMO) proves that estimation accuracy can be improved as compared to COCOMO model using industry project data [6, 7].

FL solves the problems of vagueness, imprecise and incomplete data to make reliable and accurate effort estimates. FL can be used to develop SCE model by fuzzifying functional points, applying membership functions e.g. triangular, trapezoidal, Gaussian function etc. to represent the cost drivers and defuzzifying the results to get the resultant effort. SCE model developed using FL with membership functions gives better performance as compared to COCOMO model which was tested and evaluated on a dataset of software projects [8]. Triangular fuzzy logic on NASA software projects representing linguistic terms in Function Point Analysis (FPA) with complexity metrics estimates size in person hours [9]. FL with Gaussian Membership Function (GMF) applied on COCOMO cost drivers gives results close to the actual effort than the trapezoidal function [10]. FL with Takagi-Sugeno technique for estimation applied on COCOMO and SLOC using Function Point (FP) gives simple, better estimation capabilities and mathematical relationship between the effort and the inputs [11, 12, 13].

Genetic Programming provides a more advanced mathematical function to predict more accurate estimated effort. Data mining tool can be used to increase accuracy of effort estimation by selecting a subset of highly predictive attributes such as project size, development, and environment related attributes. GA can be used to assess software project in terms of effort computation that takes much less time and performs better than COCOMO model on NASA software project dataset. GA can provide better results as compared to COCOMO II as tested on Turkish and Industry data set [14–16].

PSO with clustering can perform efficient effort estimation with learning ability by providing an efficient, flexible and user friendly way to perform the task of effort estimation. More accuracy in SCE can be achieved than the standard COCOMO using PSO with K-means clustering applied on COCOMO model that enables learning from past project data and domain specific projection of future resource requirements. PSO with inertia weight applying on COCOMO data of NASA software project can be used to calculate MARE, VARE and VAF [17, 18, 19].

Any one of the Line of Code, Function Point and Cosmic FFP can be used to measure size of a software project. Cosmic FFP provides simple, easy to use, proven and practical solution for software size estimation and quality improvement. COSMIC FFP uses functional size unit for SCE where One Cosmic Functional Size Unit (CFSU) is assigned for each entry/exit of a data group and for each read/write operation by a data group [20].

## 3 Statistical Criteria to Analyze and Evaluate Performance of SCE Model

Statistical criteria to analyze and evaluate the efficiency of software cost estimation model have been shown in Table 2.

## 4 Proposed Model

Literature analysis reveals that the SCE models developed using Neural Network, Fuzzy Logic or combination of both provides good results as compared to other soft computing techniques. Neuro-Fuzzy model acts as a powerful tool to predict cost and quality by integrating numerical data and expert knowledge. Proposed neuro-fuzzy model has been derived from [2, 6, 7, 12]. Model has been validated through data got from PROMISE Software Engineering Repository of 93 NASA projects. For calculation of effort, COCOMO II model has been used:

$$Effort = A \times (KLOC)^{B + 0.01 \times \sum_{i=1}^{5} SF_i} \times \prod_{j=1}^{17} EM_j$$

$$Schedule(inmonths) = C \times Effort^{D} + 0.01 \times \sum_{i=1}^{5} SF_i$$

where A, B, C, D are domain specific parameters (By default A = 2.94, B = 0.91, C = 3.67, D = 0.28), SF is the scale factor and EM is the effort multiplier.

Cost drivers are used in calculation of development effort, such as analyst capability, application experience etc. Fuzzification converts the crisp data to linguistic variables which are passed to Inference Engine. A fuzzy set has been defined for six qualitative rating levels for every cost driver and expressed in linguistic terms as very low (VL), low (L), nominal (N), high (H), very high (VH) and extra high (XH). The membership functions used is triangular functions which is a three-point function, defined by minimum (α), maximum (β) and modal (m) values, i.e. T (α, m, β), where (α ≤ m ≤ β). The rules can be on the basis of single parameter or combination of parameters e.g.

if (PREC is Very Low) then (EFFORT is Extra High)
if(PREC is Low) then (EFFORT is Very High)
if(FLEX is Very Low) then (EFFORT is Extra High) etc.

For defuzzification, Centeroide Method which calculates Centre of Gravity (COG) area under the curve has been used.

**Table 2** Evaluation criteria for SCE models based on actual effort and predicted effort

| S. No. | Evaluation term | Evaluation formula | Description |
|---|---|---|---|
| 1 | Relative Error (RE) | $RE(i) = \left\lvert \frac{E_P(i) - E_A(i)}{E_A(i)} \right\rvert \; where \; i = 1 \ldots n$ | RE is used to measure accuracy |
| 2 | Magnitude of Relative Errors (MRE) | $MRE = \frac{\lvert E_P - E_A \rvert}{\lvert E_A \rvert}$ | SCE model with lower MRE is better as compared to higher MRE |
| 3 | Variance Absolute Relative Error (VARE) | $VARE = Var[MRE]$ | SCE model with lower VARE is better as compared to higher VARE |
| 4 | Mean Magnitude of Relative Error (MMRE) | $MMRE = \frac{1}{n} \sum_{i=1}^{n} MRE_i$ | MMRE assesses the performance of competing models to predict accuracy but has drawback of overestimation in case of many circumstances. An effort prediction models with MMRE ≤ 0.25 is considered as acceptable SCE model with lower MMRE is better as compared to higher MMRE |
| 5 | Median of Magnitude of Relative Error (MdMRE) | $MdMRE = Median(MRE_i)$ | MdMRE is less sensitive to extreme values, while MMRE is sensitive to the outliers SCE model with lower MdMRE gives better accuracy |
| 6 | Magnitude of Error Relative to estimate (MER) | $MER = \frac{\lvert E_P - E_A \rvert}{\lvert E_P \rvert}$ | SCE model with lower MER model is better as compared to higher MER |

(continued)

**Table 2** (continued)

| S. No. | Evaluation term | Evaluation formula | Description |
|---|---|---|---|
| 7 | Mean Magnitude of Error Relative to estimate (MMER) | $MMER = \frac{1}{n}\sum_{i=1}^{n} MER_i$ | SCE model with lower MMER model is better as compared to higher MMER Accuracy of an estimation technique is inversely proportional to the MMER/MMRE if (MMRE is large and MMER is small) then average actual effort < average estimated effort. else if (MMER is large) then average estimated effort < average actual effort |
| 8 | Percentage Relative Error Deviation (PRED)—Prediction of specific Level l | $pred(l) = \frac{k}{n}$ | SCE model with higher PRED is better as compared to lower PRED as accuracy of SCE model is directly proportionally to pred(l). A prediction model is considered as acceptable when its accuracy level is 75% Pred(l) is the probability of the SCE model having relative error less than or equal to $l$ i.e. MRE ≤ $l$%; where $k$ is no. of observation and |

(continued)

**Table 2** (continued)

| S. No. | Evaluation term | Evaluation formula | Description |
|---|---|---|---|
| | | | $n$ is total no. of observations |
| 9 | Variance Account For (VAF) | $VAF(\%) = \left(1 - \frac{Var(E_A - E_P)}{Var(E_A)}\right)$ | VAF measures future outcomes likely to be predicted by the SCE model. SCE model with *higher* VAF is better as compared to *lower* VAF |
| 10 | Pearson's Correlation Coefficient (CC) | $CC(n) = \frac{\sum_{i=1}^{n}\left[\left(E_A(i) - \bar{E}_{A,n}\right)\left(E_P(i) - \bar{E}_{P,n}\right)\right]}{\sqrt{\left[\sum_{i=1}^{n}\left(E_A(i) - \bar{E}_{A,n}\right)^2\right]\left[\sum_{i=1}^{n}\left(E_P(i) - \bar{E}_{P,n}\right)^2\right]}}$ | CC between the actual and predicted estimation values indicates whether the actual and the predicted values move in the same direction. There are 3 conditions on behalf of CC: (a) $\|C\| \approx 1$ signifies a perfect estimation of the actual values by the predicted one (b) −ve CC signifies that the predicted values follow the same direction of the actual with negative mirroring i.e. with an 180° rotation about the time-axis (c) $C \approx 0$ signifies poor performance on the basis of predictions in capturing the evolution of actual values |

(continued)

**Table 2** (continued)

| S. No. | Evaluation term | Evaluation formula | Description |
|---|---|---|---|
| 11 | Root Mean Squared Error (RMSE) | $RMSE(n) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[E_P(i) - E_A(i)]^2}$ | RMSE is the square root of Mean Squared Error (MSE) that measures difference between predicted values by a SCE model and the actual values |
| 12 | Normalized Root Mean Squared Error (NRMSE) | $NRMSE(n) = \frac{RMSE(n)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[E_A(i) - \bar{E}_{A,n}\right]^2}}$ | NRMSE assess the quality of predictions using RMSE<br>(a) NRMSE = 0 signifies predictions are perfect;<br>(b) NRMSE = 1 signifies prediction is no better than taking $E_P$ equal to the mean value of n samples |
| 13 | Logarithmic Standard Deviation (LSD) | $LSD = \sqrt{\frac{\sum_{i=1}^{n}\left[(\ln E_P - \ln E_A) + \frac{s^2}{2}\right]^2}{n-1}}$<br>where $s^2$ is an estimator of the variance of the residual | S.D. provides a measure of deviation that can be expected in the final number LSD should be minimized for a good model. |

$$E = \frac{w_1(a\alpha^b) + w_2(am^b) + w_3(a\beta^b)}{w_1 + w_2 + w_3}$$

where $w_1$, $w_2$ and $w_3$ are weights of the optimistic, most likely and pessimistic estimate respectively. Here maximum weight is given for most expected estimate. ($a\alpha^b$) denotes optimistic estimate, ($am^b$) denotes most likely estimate and ($a\beta^b$) denotes pessimistic estimate.

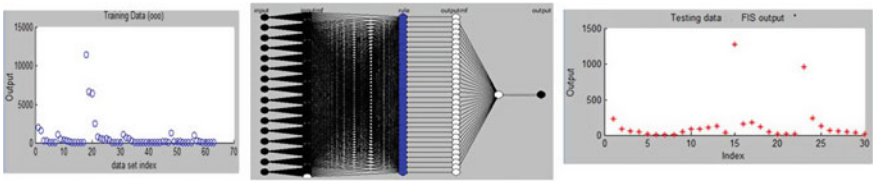


**Fig. 1** ANFIS generation using clustering of training data, model structure and testing with error 0.0058664

Proposed model has been implemented in MatLab R2013 using ANFIS (Adaptive Neuro-Fuzzy Inference System) with a hybrid learning algorithm of least-squares method and back-propagation gradient descent (for small projects) and Resilient BPNN (for large project) are used to identify parameters of Sugeno-type fuzzy inference systems as shown in Fig. 1.

## 5 Empirical Analysis and Evaluation

Hereby we are going to evaluate some popular models with our proposed model based on the statistical criteria as defined in Sect. 3. The empirical evaluations have been derived from statistical analysis of predicted effort and actual effort given by model of that particular type of technique.

## 6 Result Analysis

From the data obtained by empirical calculation of selected SCE models, it can be verified that models based on Neural Network, Fuzzy Logic or their combination performs better than other methods i.e. GA and PSO.

Based on Table 3 to find the optimized model, Table 4 and Fig. 2 reveal that for all statistical criteria, no model is at 1st rank while proposed model gives at least 2nd rank. Although the proposed Neuro-Fuzzy model does not give best results for all statistical parameters, still if we compare in overall, we can say that the proposed model provides optimized result for SCE.

**Table 3** Evaluation of techniques used in SCE model using statistical parameters

| Technique | Author/Parameter | Concept used | VARE | MMRE | MdMRE | PRED (25 %) | MMER | LSD | VAF | Corr. Coeff. | NRMSE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SCE model evaluation criteria → | | | (Min.) | (Min.) | (Min.) | ≈1 | (Min.) | (Min.) | (Max.) | ≈1 | ≈0 |
| FUZZY LOGIC | H. Mittal (2007) | Triangular | 0.008984 | 0.121702 | 0.094704 | 0.9 | 0.104432 | 0.153540 | 0.985009 | 0.993011 | 0.131985 |
| | | Triangular | 0.025013 | 0.108036 | 0.052932 | 0.9 | 0.089959 | 0.196215 | 0.991506 | 0.995744 | 0.094367 |
| | S. Reddy (2009) | Gauss. MF | 0.026421 | 0.170208 | 0.147061 | 0.8 | 0.190517 | 0.241454 | 0.900426 | 0.951539 | 0.328191 |
| | A. Mittal (2010) | Triangular | 0.129992 | 0.392957 | 0.260556 | 0.4615 | 0.496572 | 0.641570 | 0.305086 | 0.657381 | 0.874154 |
| | S. Kumar (2011) | FL Poly. Reg. | 0.035704 | 0.421978 | 0.408259 | 0.1538 | 0.742063 | 0.543016 | 0.344225 | 0.757476 | 0.892070 |
| | Ziauddin (2013) | Triangular | 0.003763 | 0.075120 | 0.074931 | 0.9666 | 0.078539 | 0.097462 | 0.987720 | 0.998464 | 0.128870 |
| | A. F. Sheta (2013) | Takagi-Sugeno | 0.180757 | 0.433817 | 0.335163 | 0.4166 | 0.347397 | 0.769945 | 0.961584 | 0.980603 | 0.192640 |
| | | FP | **0.003580** | **0.049489** | **0.021706** | **1** | **0.048550** | **0.077305** | **0.997413** | **0.998705** | **0.050862** |
| NEURAL NETWORK | G. Kumar (2014) | BPNN | 0.044214 | 0.276810 | 0.2710459 | 0.45 | 0.364313 | 0.392395 | 0.930055 | 0.967339 | 0.267792 |
| | S. Reddy (2008, 2010) | RBFN | 0.023156 | 0.240936 | 0.262962 | 0.4 | 0.330535 | 0.293232 | 0.793182 | 0.929154 | 0.532589 |
| | | BPNN | 0.077821 | 0.327361 | 0.219964 | 0.5384 | 0.373670 | 0.495869 | 0.940268 | 0.990797 | 0.250661 |
| | | RBPNN | 0.046815 | 0.302310 | 0.250113 | 0.4615 | 0.616836 | 0.445945 | 0.960052 | 0.984979 | 0.217702 |
| | A. Kaushik (2013) | BPNN | 0.004498 | 0.098403 | 0.089 | **1** | 0.114903 | 0.119214 | 0.985564 | 0.996473 | 0.155138 |
| | | BPNN | **0.003328** | **0.062864** | **0.04625** | **1** | **0.060725** | **0.084652** | **0.993602** | **0.999942** | **0.092258** |
| | A. Bawa (2012) | BPNN | 1.384785 | 1.021585 | 0.517221 | 0.3 | 0.651527 | 2.935350 | -0.274271 | 0.206923 | 1.132091 |
| | | Cascade Corr. | 0.700430 | 0.622814 | 0.374323 | 0.5 | 0.388678 | 1.650447 | 0.436012 | 0.703326 | 0.878329 |
| NEURO—FUZZY | X. Huang (2003) | all data | 0.002424 | 0.077057 | 0.058932 | **1** | 0.075707 | 0.089723 | 0.966624 | 0.995148 | 0.196549 |
| | | partial data | 0.001684 | 0.084438 | 0.068816 | **1** | 0.084578 | 0.092836 | 0.966193 | 0.994410 | 0.192589 |
| | | large weight | 0.003004 | 0.060044 | **0.036102** | **1** | 0.059734 | 0.078563 | 0.975723 | 0.995815 | 0.166063 |
| | | w/o mono. const. | **0.000723** | **0.043597** | 0.040457 | **1** | **0.042495** | **0.050128** | **0.990973** | **0.999256** | **0.082849** |
| | Proposed Model | ANFIS with BPNN, RBPNN | **0.000736** | **0.046238** | **0.030176** | **1** | **0.042359** | **0.061820** | **0.993437** | **0.999442** | **0.071682** |

(continued)

**Table 3** (continued)

| Technique | Author/Parameter | Concept used | VARE | MMRE | MdMRE | PRED (25 %) | MMER | LSD | VAF | Corr. Coeff. | NRMSE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SCE model evaluation criteria → | | | (Min.) | (Min.) | (Min.) | ≈1 | (Min.) | (Min.) | (Max.) | ≈1 | ≈0 |
| GENETIC ALGORITHMS | N. Sharma (2013) | LOC | 0.020419 | 0.232549 | 0.210933 | 0.6111 | 0.279118 | 0.286695 | 0.927928 | 0.974623 | 0.270758 |
| | | LOC + ME | 0.018985 | 0.205639 | 0.225520 | 0.6666 | 0.262881 | 0.257503 | 0.933351 | 0.966105 | 0.286109 |
| | | LOC + ME + d | **0.014125** | **0.203292** | 0.194742 | **0.7222** | **0.247370** | **0.243233** | 0.919930 | 0.964684 | 0.287741 |
| | A. F. Sheta (2006) | LOC | 0.073210 | 0.237887 | **0.145028** | 0.6111 | 0.288359 | 0.414983 | 0.970821 | 0.985310 | 0.173369 |
| | | LOC + ME | 0.329407 | 0.636396 | 0.492722 | 0.3888 | 0.322478 | 1.127379 | 0.975648 | 0.987779 | 0.272828 |
| | A. Dhiman (2013) | LOC | 0.372690 | 0.491585 | 0.185454 | 0.6 | 0.268257 | 0.960162 | **0.989669** | **0.996946** | **0.103586** |
| PSO | Hari (2011) | PSO Clustering | 0.009694 | 0.126253 | 0.105285 | 0.7777 | 0.119264 | 0.159424 | **0.983950** | **0.994124** | **0.126812** |
| | P. Reddy (2011) | PSO | **0.002565** | **0.049020** | **0.031160** | **1** | **0.049372** | **0.068906** | 0.981987 | 0.991859 | 0.134442 |
| | Sweta (2013) | Multi. Obj. PSO | 2.930059 | 0.801342 | 0.263577 | 0.4761 | 0.295084 | 7.570433 | -0.129914 | 0.585991 | 1.101915 |
| | | Sup. Vect. Reg. | 1.828605 | 0.870634 | 0.33775 | 0.3809 | 0.502089 | 3.723324 | 0.657335 | 0.831674 | 0.626645 |

**Table 4** Evaluation of SCE techniques w.r.t. statistical parameters

| Technique/Parameter | Neural N/w | Fuzzy logic | Neuro-fuzzy | Neuro-fuzzy |
|---|---|---|---|---|
| | A. Kaushik (2013) | A.F. Sheta (2013) | X. Huang (2003, 2007) | Proposed model |
| **NRMSE (≈0)** | 0.089353 | **0.051238** | 0.083758 | 0.071682 |
| **VARE** (Min) | 0.002347 | 0.004136 | **0.000613** | 0.000736 |
| **MMRE** (Min) | 0.073145 | 0.052658 | **0.042164** | 0.046238 |
| **MdMRE** (Min) | 0.043192 | **0.023515** | 0.041347 | 0.030176 |
| **MMER** (Min) | 0.071634 | 0.049124 | **0.043373** | **0.042359** |
| **LSD** (Min) | 0.096564 | 0.076632 | **0.061282** | 0.06182 |
| **VAF** (Max) | 0.991533 | **0.996134** | 0.991763 | 0.993437 |
| **Corr. Coeff. (≈1)** | **0.999856** | 0.998615 | 0.999467 | 0.999442 |
| **PRED (25 %) (≈1)** | **1** | **1** | **1** | **1** |

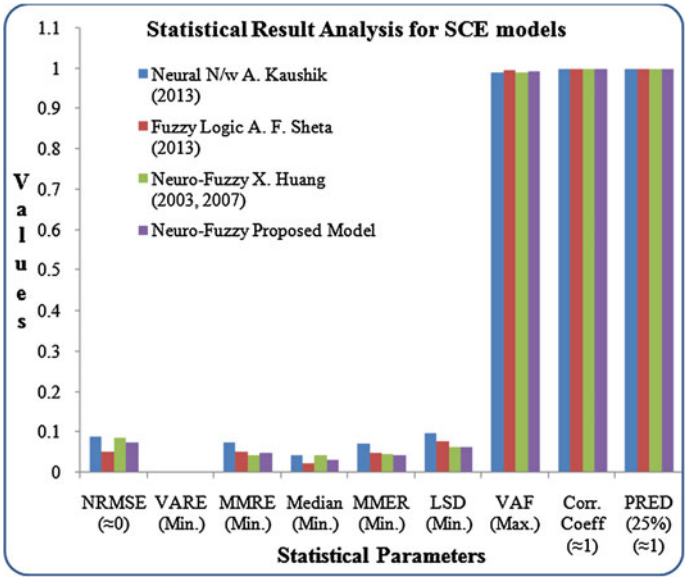


**Fig. 2** Performance of Proposed model with other considered SCE models

## 7 Conclusion and Future Works

In this paper, a detailed empirical analysis and evaluation of SCE models developed through soft computing techniques (e.g. ANN, FL, GA, PSO etc.) have been done using an in-depth review and statistical criteria. Final results indicates that none of the models shows perfect behavior as in terms of certain measure, certain model qualifies better than another, but for other measures it may be worse. Analytical

review of considered models shows that SCE models based on NN, FL or combination of NN and FL can give better results as compared to SCE models based on other techniques. Keeping this view, an optimized Neuro-Fuzzy SCE model has been proposed that provides optimum results for considered statistical parameters as compared to other considered SCE models. Due to limitations of NN and FL, proposed model is dependent on size and type of project and data used for training/learning. As per the empirical analysis, it seems that while developing SCE model there is still some scope of improvement in Neuro-Fuzzy techniques and can be accommodated in near future. In future, some improvements may be done by developing SCE models using other optimization techniques like Ant Colony Optimization, Bee Colony Optimization etc. to overtake the performance given by proposed Neuro-Fuzzy SCE model.

## References

1. Kumar, G., Bhatia, P.K.: Automation of software cost estimation using neural network technique. Int. J. Comput. Appl. **98**(20), 11–17 (2014)
2. Kaushik, A., Soni, A.K., Soni, R.: A simple neural network approach to software cost estimation. Global J. Comput. Sci. Technol. **13**(1), Version 1, 23–30 (2013)
3. Bawa, A., Chawla, R.: Experimental analysis of effort estimation using artificial neural network. Int. J. Electron. Comput. Sci. Eng. **1**(3), 1817–1824 (2012)
4. Reddy, C.S., Raju, K.: An optimal neural network model for software effort estimation. Int. J. Softw. Eng. **3**(1), 63–78 (2010)
5. Reddy, C.S., Sankara Rao, P., Raju, K., Valli Kumari, V.: A new approach for estimating software effort using RBFN network. Int. J. Comput. Sci. Netw. Secur. **8**(7), 237–241 (2008)
6. Huang, X., Ho, D., Ren, J., Capretz, L.F.: Improving the COCOMO model using a neuro-fuzzy approach. Elsevier J. Appl. Soft Comput. **7**, 29–40 (2007)
7. Huang, X., Capretz, L.F., Ren, J., Ho, D.: A neuro-fuzzy model for software cost estimation. In: Proceedings of the IEEE 3rd International Conference on Quality Software, 126–133, 6–7 Nov 2003
8. Mittal, A., Parkash, K., Mittal, H.: Software cost estimation using fuzzy logic. ACM SIGSOFT Softw. Eng. Notes **35**(1), 1–7 (2010)
9. Mittal, H., Bhatia, P., Optimization criteria for effort estimation using fuzzy technique. CLEI Electron. J. **10**(1), Paper 2, 1–11 (2007)
10. Reddy, C.S., Raju, K., An improved fuzzy approach for COCOMO's effort estimation using gaussian membership function. J. Softw. **4**(5), 452–459 (2009)
11. Ziauddin, K.S., Khan, S., Nasir, A.J.: A fuzzy logic based software cost estimation model. Int. J. Softw. Eng. Appl. **7**(2), 7–17 (2013)
12. Sheta, A.F., Aljahdali, S.: Software effort estimation inspired by COCOMO and FP models: a fuzzy logic approach. Int. J. Adv. Comput. Sci. Appl. **4**(11), 192–197 (2013)
13. Swarup Kumar, J.N.V.R., Mandala, A., Vishnu Chaitanya, M., Prasad, G.V.S.N.R.V., Fuzzy logic for software effort estimation using polynomial regression as firing interval. Int. J. Comput. Technol. Appl. **2**(6), 1843–1847 (2011)
14. Sharma, N., Sinhal, A., Verma, B.: Software assessment parameter optimization using genetic algorithm. Int. J. Comput. Appl. **72**(7), 8–13 (2013)
15. Sheta, A.F.: Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects. J. Comput. Sci. **2**(2), 118–123 (2006)

16. Dhiman, A., Diwaker, C.: Optimization of COCOMO II effort estimation using genetic algorithm. Am. Int. J. Res. Sci. Technol. Eng. Math. 208–212 (2013)
17. Hari, C.V.M.K., Sethi, T.S., Jagadeesh, M.: SEEPC: A toolbox for software effort estimation using soft computing techniques. Int. J. Comput. Appl. **31**(4), 12–19 (2011)
18. Prasad Reddy P.V.G.D., Hari, C.V.M.K.: Software effort estimation using particle swarm optimization with inertia weight. Int. J. Softw. Eng. (IJSE) **2**(4), 87–96 (2011)
19. Kumari, S., Pushkar, S.: Comparison and analysis of different software cost estimation methods. Int. J. Adv. Comput. Sci. Appl. **4**(1), 153–157 (2013)
20. Kumar, G., Bhatia, P.K.: A detailed analysis of software cost estimation using COSMIC-FFP. PAK Publishing Group J. Rev Comput. Eng. Res. **2**(2), 39–46 (2015)
21. Kaushik, A., Chauhan, A., Mittal, D., Gupta, S.: COCOMO estimates using neural networks. Int. J. Intell. Syst. Appl. **9**, 22–28 (2012)