Artificial Neural Networks Based Analysis of Software Cost Estimation Models

Farhad Soleimanian Gharehchopogh[1], Isa Maleki[2], Sahar Sadouni[3]

[1]Department of Computer Engineering, Hacettepe University, Beytepe, Ankara, Turkey
[2]Young Researchers and Elite Club, Urmia Branch, Islamic Azad University, Urmia, Iran
[3]Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

Abstract: *Managers required administering software development projects with accurate estimates of the resources. So accurate estimates can increase the speed of the effort for developing software projects, and prevent probabilistic failure consequently. Therefore, in the process of software development estimate must be taken in order to reduce cost and time schedule and the existence of probabilistic risks. The implementation of software projects developmental process in large organizations with hundreds of experts and professionals often is a complicated action and needs to be estimated. Estimation cost, time and manpower depends to factors such as the performance of software groups and the complexity of software project. Different models have been proposed for the Software Cost Estimation (SCE) that COCOMO model is the most common model for SCE. This model was used in 1981 by Boehm. This model is a high risk and threat for software projects due to its low accuracy and lack of reliability. Therefore, there is a need to evaluate and estimate SCE using Artificial Intelligence (AI) models. One of the best models of AI for SCE is Artificial Neural Networks (ANNs). ANNs try to assess and evaluate the data with minimal error using instruction and learning techniques. In this paper, the performance of Multi-Layer Perceptron (MLP), Radial Basis Function (RBF), Wavelet Neural Network (WNN), Functional Link Artificial Neural Network (FLANN) and Generalized Regression Neural Network (GRNN) models have been discussed in SCE.*

Keywords: *Software Cost Estimation, COCOMO, Artificial Neural Networks*

1. Introduction

One of the main reasons which lead to software project development's failure is inaccurate estimation of the cost and time. Therefore, accurate estimate is the potential factor in estimation of cost, time and effort. The important criterion in a software project is the Source Line of Code (SLOC). Therefore, software project complexity is a function of the SLOC. In the development of software projects must be careful because cost and effort factors affect in the designing and implantation and may lead to quality problems and cost increases [1]. In addition to cost, another important factor in the development of software projects is quality. So, if the software had not enough quality, its development and maintenance would be costly and this lead to the failure of the software and the extra cost, besides, the software would not have the required performance. Therefore, to estimation the cost and effort in software projects, different algorithmic software models such as COCOMO I [9], COCOMO II [7], SLIM [11, 12] and FP [3] has been used.

One of the topics covered during the past 30 years and devoted many researches is cost estimation for the software projects. Important issues in the field of software engineering are the capability of cost estimation and required factors affecting to the development of software projects. Inaccurate cost estimate often leads to projects failure and undesirable results for the project development teams. Since the software project managers choose the best method of estimation based on condition and project descriptions, the description and the composition of the estimation techniques can be useful for reducing project failure [16, 25]. It cannot be said that there are certain ways to estimate the entire projects accurately, but each technique is appropriate for a particular project. Because the performance of each method depends on several parameters, including project complexity, project duration, staff expertise, development methodology, and so on. One of the most important steps in the development of software is ECS. In software development, lots of project's parts analyzed manually by human factor. Since the possibility of human error in this procedure is high and it is likely to be threatened by the development of software, then there will be the possibility of not testing influential factors in cost estimating of software projects. Therefore, greater automation and intelligent software development projects lead to

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.75)

producing software with lower cost and time. Evaluation of factors and affecting process often require an accurate, proper and comprehensive estimation of the model in the development of software projects [22, 23 and 24]. For presentation of such model, all parameters in the process of software development and simultaneous effect of these factors on the output software projects should be accurately determined. In algorithmic models that use multiple inputs and outputs, determining influential factors in the estimation is difficult. In recent years, ANNs model of the SCE are given acceptable results and in many cases they have been much more accurate models of algorithms [20, 15].

The structure of the paper is as follows: In Section 2, we'll deal with evaluation criteria in SCE; in Section 3, we will review ANNs are models in the SCE, in Section 4, we will discuss ANNs models in the SCE, and finally in Section 5, we'll deal with the conclusions and future works.

2. Evaluation Criteria in SCE

Equations of evaluations criteria show increased accuracy and realistic estimation by the estimation models. Since the purpose of criteria implementation is the identification of accurate and precise operations of algorithms estimation. In Table (1) the most important evaluation criteria in SCE is shown.

Table 1. The most important evaluation criteria in SCE

Evaluation Criteria	Description
$RE_i = \frac{\lvert act_i - est_i \rvert}{act_i}$	Relative Error (RE). Value RE obtained from the actual estimated minus estimate different models.
$MRE_i = RE_i \times 100$	Magnitude of Relative Error (MRE). An important criterion for the evaluation of cost estimation with other models is the MRE. [10].
$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i$	Mean MRE (MMRE). MMRE is a common method used for evaluation prediction models [19].
$MdMRE = Median(MRE)$	Median MRE (MdMRE). MdMRE is criterion for mean MRE error. MdMRE has been used as another criterion because it is less sensitive to outliers [10].
$MER_i = \frac{\lvert act_i - est_i \rvert}{est_i} \times 100$	Magnitude Error Relative (MER). MER measures the error relative to the estimate [10].
$MMER = \frac{1}{N}\sum_{i=1}^{N} MER_i$	Mean of MER (MMER). The aggregation of MER from multiple observations (N) can be achieved through the mean of MER.
$MAE = \frac{1}{n}\sum_{i=1}^{n} \lvert act_i - est_i \rvert$	Mean of Absolute Errors (MAE). MAE evaluations the estimates are from actual values in projects number. It could be applied to any two pairs of numbers, where one set is "Actual" and the other is an estimate prediction.
$MAPE = \sum_{i=1}^{n}(\frac{\lvert act_i - est_i \rvert}{act_i}) / n \times 100$	Mean Absolute Percentage Error (MAPE). MAPE of the estimated values with respect to the actual amount of the development effort-hour for each project was calculated [10].
$MSE = \frac{1}{n}\sum_{i=1}^{n}(act_i - est_i)^2$	Mean Squared Error (MSE). MSE is the mean of the square of the differences between the actual and the predicted efforts [10].
$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(act_i - est_i)^2}$	Root Mean Square Error (RMSE) is frequently used evaluation of differences between values estimated by a model or estimator and the values actually observed from the thing being modelled or estimated [19].

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.75)

$PRED(x) = \frac{1}{n} \times \sum_{i=1}^{n} \begin{cases} 1, & ifMRE \le x \\ 0, & otherwise \end{cases}$	Where n denotes the total number of projects and k denotes the number of projects whose MRE is less than or equal to q. normally, q is set to be 0.25 [19].
$CC = \frac{\sum_{i=1}^{n}(act_i - \overline{act})(est_i - \overline{est})}{\sqrt{\sum_{i=1}^{n}(act_i - \overline{act})^2 \sum_{i=1}^{n}(est_i - \overline{est})^2}}$	Correlation Coefficient (CC). Correlation measures of the strength of the relationship between two variables. The strength of the relationship is indicated by the CC. The larger the value of CC, the stronger the relationship [19].

3. ANNs for SCE

ANNs consists of a collection of interconnected neurons that each set of neurons is called a layer [29, 14]. ANNs are composed of an input layer, one or more hidden layers and an output layer. The shape and the type of neuronal connectivity in different layers are causing different structures of ANNs. In Figure (1) the SCE model using ANNs is shown.

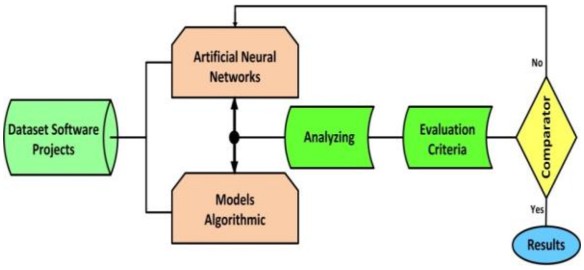

Figure. 1. SCE Model using ANNs

Many attempts in recent decades are done to develop and improve SCE models using ANNs. One obvious advantage of such a model to other nonlinear models is that ANNs are smart estimators which can estimation any function with optional precision. There is no need to recognize the particular model in ANNs and the model's forms based on the information in the data.

MLP and RBF networks are used for SCE [17]. Evaluation has been carried out on KEMERER dataset with 15 projects, IBMDPS with 24 projects and Hallmark with 28 projects. Table (2) shows MAPE criterion for three given dataset using MLP and RBF networks. Also, for the sake of showing the performance of the MLP and RBF networks, experimental results were compared with regression models. The results indicate that MAPE criterion with MLP and RBF networks have better accuracy in comparison with Regression model.

Table 2. Evaluation of Criterion MAPE using MLP and RBF

Actual Effort	Models [17]			
	ANNs		Regression Analysis	
	Estimated Effort FP/RBF	Estimated Effort LOC/MLP	Estimated Effort FP	Estimated Effort LOC
11.42	26.75	9.55	29.95	7.99
13.14	10.90	15.85	17.78	12.03
23.30	32.45	25.32	32.36	22.22
38.10	38.74	19.54	30.84	15.99
61.20	63.38	22.30	41.56	19.05
3.60	7.85	3.72	10.09	2.08
11.80	9.52	4.43	15.13	3.50
MAPE	47.60	31.96	70.86	40.37

MLP and the COCOMO II hybrid models are used for SCE [28]. Evaluation was performed on COCOMO 81 dataset. The main goal of this combination is the better education of input data by the MLP. In the combinational models, Effort Multiplier (EM) and Scale Factor (SF) factors are taught by a middle layer

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.75)

and are tested and evaluated in COCOMO II model defined by equation (1).

$$PM = A.(Size)^{1.01+\sum_{i=1}^{5} SF_i} . \prod_{i=1}^{17} EM_i \qquad (1)$$

Combining the COCOMO II and MLP models can improve the weakness of the COCOMO II model using the strengths of the MLP model. Input layer in MLP consists of 5 nodes in SF, 17 nodes in EM and 2 nodes in Bias. For the middle layer activation function, sigmoid function has been used according to equation (2). Also back-propagation is used for training the middle layer. Experimental results show that the MLP models error is greatly reduced using MMRE.

$$f(x) = \frac{1}{1-e^{-x}} \qquad (2)$$

The combination of clustering algorithms C-Means [8] and APC-III [30] with RBF networks is used for SCE [17]. For evaluation, software projects dataset COCOMO 81 has been used which includes 252 projects. For the middle layer activation function, the Gaussian function has been used according to equation (3). The results show that the hybrid models C-Means and RBF have the better performance in project clustering in comparison with APC-III hybrid model and RBF and they reduced MMRE criterion.

$$f(x) = e^{(-\frac{\|x-c_i\|^2}{\sigma_i^2})} \qquad (3)$$

WNN is used for SCE [21]. Morlet function and Gaussian function have been used in WNN for the activation of the middle layer. The training and test phases of data are 80% and 20% respectively. Evaluation has been made on IBMDPS software project. For showing the performance of WNN model in SCE, it is compared with the MLP, RBF, Multiple Linear Regression (MLR), Neuro-Fuzzy Inference System (DENFIS) and Support Vector Machine (SVM) models. Table (3) shows the comparison of the models. It is observed that the WNN-Morlet and WNN-Gaussian models have less error in comparison with other models.

Table 3. Evaluation of Criterion MMRE

Models [21]	**MMRE**
MLP	1.82
MLR	3.28
RBF	0.36
DENFIS	3.28
SVM	0.92
WNN-Morlet	0.11
WNN-Gaussian	0.03

The combination of MLP and COCOMO II networks is used for SCE [27]. The evaluation has been made on 63 projects from COCOMO 81 software projects. The main objective of this compound is the better training of input data by the MLP. In the combination model, EM and SF factors is trained by the middle layer, and are tested and evaluated in the COCOMO II model. An input layer in MLP consists of 5 SF nodes, 17 EM nodes and 2 bias nodes. For the activation function of the intermediate layer, the sigmoid function is used. Also for training of the middle layer, back-propagation is used. The training and test phases of data are 80% and 20% respectively. Test results showed that the standard error of MRE in a hybrid model is less than the COCOMO II model.

A new approach has been proposed for the SCE using a combination of MLP and COCOMO II [2]. Evaluation of two dataset has been carried out with 63 projects and 100 projects. The main objective of this compound is better training of input data by the MLP. In the compound model, EM and SF factors are trained by the middle layer and are tested and evaluated in COCOMO II model. The input layer in MLP consists of 5 SF nodes, 17 EM nodes and 2 bias nodes. Sigmoid function is used for activation of intermediate layer's function. Also back-propagation is used for training of the middle layer. The results of the experiments in Table (4) show that the hybrid model in comparison with COCOMO II model has less error. As you can see, the criteria MMRE and PRED in MLP-COCOMO II model have lower error rate in comparison with COCOMO II model.

Table 4. Evaluation of Criteria MMRE and PRED

No. Projects	**Models [2]**	**MMRE**	**PRED(0.25)**
63	COCOMO II	0.58	30%
	MLP-COCOMO II	0.41	40%
100	COCOMO II	0.48	40%
	MLP-COCOMO II	0.46	50%
163	COCOMO II	0.53	35%
	MLP-COCOMO II	0.44	45%

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.75)

RBF and GRNN models were used for SCE [26]. Evaluation of models on COCOMO 81 dataset has been carried out with 63 projects. The results of the experiments in Table (5) show that RBF model has less errors and high accuracy in comparison with GRNN and Intermediate COCOMO models.

Table 5. Evaluation of Criteria in MMRE and PRED

Models [26]	MMRE	PRED(0.40)
Intermediate COCOMO	18.60	87.3
RBF	17.29	90.48
GRNN	34.61	84.13

MLP hybrid model and GA algorithm are used for SCE [4]. The GA algorithm is used in two hybrid models. OFWFLANN model is used in order to optimize factors affecting the cost and OCFWFLANN model are used in order to train the weight of MLP vectors. The GA fitness function will be minimized the error according to equation (4).

$$f = \frac{1}{MMRE} \tag{4}$$

Also, back-propagation is used for training of middle layer in MLP. Initial population and number of generations in GA are 2000 and 200 respectively. Evaluation has been done on NASA93 dataset. Table (6) shows the results of the experiments of GA-MLP hybrid model. Training and testing has been done on MMRE, MdMRE and PRED (0.25) criteria.

Table 6. Evaluation of Criteria MMRE, MdMRE and PRED

Models [4]	MMRE		MdMRE		PRED(0.25)	
	Training	Testing	Training	Testing	Training	Testing
OCFWFLANN	0.33	0.32	0.29	0.24	0.36	0.31
OFWFLANN	0.43	0.38	0.44	0.33	0.35	0.43
MLP	0.48	0.42	0.42	0.38	0.51	0.44

MLP network is used for SCE [20]. Evaluation is done on 63 projects from NASA software project dataset. In order to show the performance of MLP, Linear Regression (LR), Support Vector Regression (SVR) and K-Nearest Neighbors (KNN) used and the rate of errors compared in MAE, RMSE and CC. Table (7) shows the performance of given models. As you can see, ANN model has low rate of error when it is compared with the LR and the SVR models.

Table 7. Evaluation of Criteria MAE, RMSE and CC

Models [20]	MAE	RMSE	CC
LR	39.17	81.52	%99
SVR	36.32	41.09	%97
KNN	1.12	5.44	%1
ANN	12.07	16.64	%99

Hybrid model of FLANN and GA were used for SCE [5]. The GA fitness function will be minimized the error according to equation (5).

$$f = \frac{1}{MMRE} \tag{5}$$

Hybrid model of GA-FLANN is a kind of three-layer Feed Forward Network. GA algorithm is used for both hybrid models of OFWFLANN and OCFWFLANN. OFWFLANN model is used to optimize the factors affecting the cost and OCFWFLANN model is used to train the weight of FLANN vectors. Evaluation has been done on NASA93 dataset. Table (8) the results of the hybrid model of GA-FLANN shows.

Table 8. Evaluation of Criteria MMRE, MdMRE and PRED

Models [5]	MMRE		MdMRE		PRED(0.25)	
	Training	Testing	Training	Testing	Training	Testing
OCFWFLANN	0.28	0.27	0.24	0.19	0.31	0.26
OFWFLANN	0.38	0.33	0.39	0.28	0.30	0.38
FLANN	0.43	0.37	0.37	0.33	0.46	0.39

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.75)

A case study is proposed for the SCE based on the MLP network [15].The results show that ANNs model is one of the best models for SCE. Evaluations of 63 projects have been conducted on the NASA dataset. The results show that ANNs training in more than 90% presents better estimates and has less MRE error. Investigations from the ANNs test show that techniques of AI have better performance in comparison with algorithmic models and estimate the cost with higher accuracy. Despite the approximate estimation error for both methods, it should be noted that in general, using AI methods, the error will be very small.

The combination of FLANN and PSO algorithm is used for SCE [6]. Hybrid PSO-FLANN model is a kind of neural network's three-layer Feed Forward. PSO algorithm is used to train the weight of FLANN vector. Evaluation has been made on three dataset COCOMO 81, NASA63 and Maxwell. The results of the experiments of PSO-FLANN model in Table (9) is shows.

Table 9. Evaluation of Criteria MMRE, MdMRE and PRED

Dataset	MMRE		MdMRE		PRED(25)	
	Training	Testing	Training	Testing	Training	Testing
COCOMO 81	0.43	0.37	0.48	0.42	0.39	0.52
NASA63	0.49	0.34	0.44	0.45	0.39	0.50
Maxwell	0.55	0.38	0.49	0.42	0.32	0.48

The combination of MLP and COCOMO II is used for SCE [13]. Also in order to have a better education, PSO algorithm is used in the MLP middle layer. The sigmoid function is used to activate the function. Evaluation on NASA93 and COCOMO 81 dataset has been made. Table (10) shows the results of the experiments of hybrid models. It can be seen that the PSO-MLP-COCOMO II hybrid model has less MRE error in comparison with the MLP-COCOMO II model.

Table 10. Evaluation of Criteria MMRE and PRED

Dataset	Models [13]	Evaluation	
		MRE	PRED(0.25)
NASA93	MLP-COCOMO II	0.48	52%
	PSO-MLP-COCOMO II	0.45	55.1%
COCOMO 81	MLP-COCOMO II	0.42	39%
	PSO-MLP-COCOMO II	0.40	43.2%
Mean	MLP-COCOMO II	0.45	45.5%
	PSO-MLP-COCOMO II	0.43	49.5%

4. Discussion

ANNs are capable of determining the relationship between inputs and outputs in the system data which are connected to each other in a network of nodes and the amount of activity of each of these connections regulated by the training process and finally the model will be able to discover the related rules of input and output, even if the rules are non-linear and complex. In fact, ANNs are learnable and have the knowledge to solve the problems through learning ways. Learning abilities of ANNs is done using the network parameters adjustments. With this aim that if small changes occurred in environmental conditions of network, the network also be efficient to new conditions with little training. In ANNs each neuron operates independently and the behavior of the network is the result of the behavior of multiple neurons. In other words, the neurons in a process of working together correct and train each other that this character enhances the network's fault-tolerant. Unlike the algorithmic approaches, ANNs are models act according to comparative analysis and inference. Often the process of estimation in these models is done according to the analysis of training dataset.

Algorithmic models work according to mathematics and some empirical equations. The models are often difficult to learn and they need a lot of data about the current project status. However, if sufficient information is available, these models can almost give reliable results. Moreover, the algorithm approaches are complementary. Usually COCOMO used SLOC and FP as two entry criteria and generally if these two criteria be accurate, COCOMO can also

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.75)

provide accurate results. Finally, to select the best method for estimating, compliance information available to current projects and data from previous projects is helpful. Because there are uncertainties in the properties of ANNs, using these methods in all cases and per each model may not provide the desired effect, and network encounters with some percent of the modeling error. For modeling and testing software projects by ANNs should be meticulous in the aspect of selecting educational functions. Also, the most important feature of the ANNs is that they are not dependent on the input data values; this means that input data can have any arbitrary statistical distribution. This important feature of ANNs is privilege against algorithmic models and it gives them the ability to use the same from different types of input data with any desired distribution. In Table (11) comparisons models ANNs for SCE is shown.

Table 11. Comparison of ANNs on Different Dataset Software Projects

Approaches	Models	Dataset(s)	No. Projects	Evaluation Criteria
[17], [18], [26]	RBF	Kemerer	15	MAPE, MMRE, PRED(0.40)
		IBMDPS	24	
		Hallmark	28	
		COCOMO 81	252	
		COCOMO 81	63	
[20], [15], [17], [28], [30], [27], [2], [4], [13],	MLP	COCOMO 81	63	MMRE, MRE, PRED(0.25), MdMRE, CC,MAE RMSE
		Kemerer	15	
		IBMDPS	24	
		Hallmark	28	
		NASA93	93	
		NASA63	63	
[21]	WNN	IBMDPS	24	MMRE
[26]	GRNN	COCOMO 81	63	MMRE, PRED(0.40)
[4], [5]	FLANN	NASA93	93	MMRE, MdMRE, PRED(0.25)
		COCOMO 81	63	
		NASA63	63	
		Maxwell	100	

ANNs require a series of input and output data for designing and training in order to explore nonlinear or uncertain relations between them with logical analysis of these data, for example, could do the simulation for the same probabilistic items. In the papers that have been investigated, COCOMO 81, NASA (63), NASA (93), IBMDPS, Kemerer, Hallmark and Maxwell dataset used for the SCE.

Results of Evaluation of ANNs models show that MLP ANNs with sigmoid transfer function has greater accuracy and the ability in comparison with RBF neural networks in SCE. But, it should be said that finding the number of hidden layer and transfer fit function in each of ANNs are models requires trial and error, so that in each test the evaluation criteria of estimation to be repeated so long that minimizes the error between the predicted parameters obtained with the actual parameters. How to give input to the ANNs in SCE is important because the software projects dataset has many factors. Therefore, these values should be altered so as to avoid divergence in network. For easy and fast training of ANNs, it is necessary to exclude some of the data that are outside the scope of conventional data from the network's training. Therefore, reducing the scale of the problem is also a very useful method for rapid convergence of the network's models.

The most important ANNs model is RBF. This network has become one of the most famous ANNs and is considered the most important MLP competitor according to a variety of applications. This type of network requires more neurons, but its benefits are low design time in comparison with the MLP network. These networks when there are plenty of educational vectors, has the best performance. In the RBF model, the Gaussian function is used for training the network to achieve better results. Also repeating the algorithm prevents trapping in a local minimum.

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.75)

In general the results of evaluations represent the optimal performance of GRNN network in estimating the value of MMRE error in comparison with FLANN network. Although the hybrid algorithm have been used in FLANN networks to increase the accuracy of the vector parameters input, but the GRNN networks are the appropriate method for SCE than algorithmic methods and FLANN.

ANNs have less error in estimation, and the reason can be attributed to being independent on the characteristics of the data and relationships between them, because the relationship between variables may be non-linear relationship that the algorithmic models cannot predict it. In ANNs, the selection of the number of variables as network's input depends on dataset, so if the goal of estimation is the accuracy of estimation, the models should be used that have most of the input. But if the purpose of estimation be the less cost, then those variables should be used that need less time, facilities and expertise for estimation.

5. Conclusions and Future Works

SCE is one of the most challenging managerial works because many of cost factors are variable and in the early stages of software development are not also easily estimated. In this paper we saw that ANNs has a high capacity in SCE in comparison with other algorithmic models. However, due to the lack of complex formula and dynamic nature of the ANNs, they are considered a good idea for the estimation. Since the results of the models such as COCOMO in software development are time consuming and based on the assumption of hypothesis and guessing and finding the appropriate and logistic model for the given dataset is difficult, using ANNs to estimate true and accurate estimates is optimal. Based on investigations and considerations in this paper, it can be said that ANNs can be a useful tool for SCE and it can be analyzed and estimated its various software projects with large and small dataset. With consideration of ANNs in SCE, we hope to use the combination of ANNs are model for better accuracy in SCE in the future.

References

1. Araujo R. de A., Soares S., Oliveira A.L.I., “Hybrid Morphological Methodology for Software Development Cost Estimation”, Expert Systems with Applications, Vol. 39, pp. 6129-6139, 2012.

2. Attarzadeh I., Ow S.H., “Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks”, 2nd International Conference on Computer Engineering and Technology (ICCET), Chengdu, Vol. 3, pp. 487-491, 2010.

3. Albrecht A.J., Gaffney J., “Software Function, Source Lines of Code, and Development Effort Prediction: a software science validation”, IEEE Transactions on Software Engineering SE, 9(6): 639-648, 1983.

4. Benala T.R., Chinnababu K., Mall R., Dehuri S., “A Particle Swarm Optimized Functional Link Artificial Neural Network (PSO-FLANN) in Software Cost Estimation”, Advances in Intelligent Systems and Computing Proceedings of the International Conference on Frontiers of Intelligent Computing, pp. 59-66, Springer-Verlag, 2013.

5. Benala T.R., Dehuri S., “Genetic Algorithm for Optimizing Functional Link Artificial Neural Network Based Software Cost Estimation”, Proceedings of the InConINDIA, pp. 75-82, Springer-Verlag, 2012.

6. Benala T.R., Dehuri S., “Genetic Algorithm for Optimizing Neural Network Based Software Cost Estimation”, SEMCCO 2011, Part I, LNCS 7076, pp. 233-239, Springer-Verlag, 2011.

7. Boehm B.W., Software Cost Estimation with COCOMO II, Prentice Hall PTR, Englewood Cliffs, New Jersy, 2000.

8. Bezdek J. C., Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, 1981.

9. Boehm B.W., Software Engineering Economics, Prentice-Hall, Englewood Cliffs, New Jersy, 1981.

10. Capretz L.F., Marza V., Improving Effort Estimation by Voting Software Estimation Models, Hindawi Publishing Corporation, Advances in Software Engineering, pp. 1-8, 2009.

11. Cot V.P., Oligny B., Rivard N., Software Metrics: an Overview of Recent Results, the Journal of Systems and Software, Vol. 8, pp. 121-131, 1988.

12. Conte S.D., Dunsmore H.E., Shen V.Y., “Software Engineering Metrics and Models”, The Benjamin/Cummings Publishing Company, Inc, Menlo Park, CA, p. 214, 1986.

13. Dan Z., “Improving the Accuracy in Software Effort Estimation”, IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), pp. 180-186, Dongguan, 2013.

14.Engelbrecht A. P., "Computational Intelligence An Introduction", Second Edition, pp. 15-25, John Wiley & Sons Ltd, 2007.

15.Gharehchopogh F.S., "Neural Networks Application in Software Cost Estimation: A Case Study", 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011), pp. 69-73, IEEE, Istanbul, Turkey, 15-18 June 2011.

16. Gharehchopogh. F.S, Ebrahimi. L, Maleki. I., Gourabi. S.J, A Novel PSO based Approach with Hybrid of Fuzzy C-Means and Learning Automata in Software Cost Estimation, Indian Journal of Science and Technology, 7(6)795-803, 2014.

17.Heiat, A. Comparison of Artificial Neural Network and Regression Models for Estimating Software Development Effort, Information and Software Technology, Vol. 44, pp. 911-922, Elsevier Science B.V, 2002.

18.Idri A., Abran A., Mbarki S., An Experiment on the Design of Radial Basis Function Neural Networks for Software Cost Estimation", Information and Communication Technologies (ICTTA '06), Damascus, Vol. 1, pp. 1612-1617, IEEE, 2006.

19.Kumari, S. Pushkar, S. Performance Analysis of the Software Cost Estimation Methods: A Review, International Journal of Advanced Research in Computer Science and Software Engineering, 3(7): 229-238, 2013.

20.Khalifelu Z.A., Gharehchopogh F.S., Comparison and Evaluation Data Mining Techniques with Algorithmic Models in Software Cost Estimation", Elsevier, Procedia-Technology Journal, Vol. 1, pp. 65-71, 2012.

21. Kumar K.V., Ravi V., Carr M., Kiran N.R., Software Development Cost Estimation using Wavelet Neural Networks, The Journal of Systems and Software, Vol. 81, pp. 1853-1867, Elsevier Inc., 2008.

22. Maleki I., Ghaffari A., Masdari M., A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization, International Journal of Innovation and Applied Studies, 5(1): 72-81, 2014.

23. Maleki, I. Ebrahimi, L. Gharehchopogh, F.S. A Hybrid Approach of Firefly and Genetic Algorithms in Software Cost Estimation, MAGNT Research Report, 2(6): 372-388, 2014.

24. Maleki, I. Gharehchopogh, F.S. Ayat, Z., Ebrahimi, L. A Novel Hybrid Model of Scatter Search and Genetic Algorithms for Software Cost Estimation, MAGNT Research Report, 2(6): 359-371, 2014.

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.75)

25. Maleki. I, Ebrahimi. L, Jodati. S, Ramesh. I, Analysis of Software Cost Estimation Using Fuzzy Logic, International Journal in Foundations of Computer Science & Technology (IJFCST), 4(3) 27-41, 2014.

26.P. Reddy P.V.G.D, S.K.R, R. Sree P, Ramesh S.N.S.V.S.C, Software Effort Estimation using Radial Basis and Generalized Regression Neural Networks, Journal of Computing, 2(5): 87-92, 2010.

27.Reddy, C.S. KVSVN, Raju. A Concise Neural Network Model for Estimating Software Effort, International Journal of Recent Trends in Engineering, 1(1): 188-193, 2009.

28.Tadayon N., Neural Network Approach for Software Cost Estimation, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), Vol. 2, pp. 815-818, 2005.

29.Wena J., Li S., Lin Z., Hu Y., Huang C., Systematic Literature Review of Machine Learning based Software Development Effort Estimation Models, Information and Software Technology, Vol. 54, pp. 41-59, 2012.

30. Y-S, Hwang. S-Y, Bang. An Efficient Method to Construct a Radial Basis Function Network Classifier, Neural Networks, 10(8): 1495-1503, 1997.