International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 12665-12681


# Review on Intelligent and Soft Computing Techniques to Predict Software Cost Estimation

**Venkataiah V**
*Associate Professor, Computer Science and Engineering,*
*CMR College of Engineering & Technology, Medchal, Hyderabad-500055, India.*
*Orcid: 0000-0002-7999-4866*

**Ramakanta Mohanty**
*Professor, Computer Science and Engineering,*
*Keshav Memorial Institute of Technology, Narayaguda, Hyderabad-500011, India.*

**Nagaratna M**
*Assistant Professor, Computer Science and Engineering,*
*Jawaharlal Nehru Technological University Hyderabad- College of Engineering,*
*Kukatpally, Hyderabad-500085, India.*

**Abstract**

This article, presents the Machine Learning, Intelligent and Soft computing techniques were applied to predict the estimated cost, quality, accuracy of software developed during 2000-2017. This article, gives a 95% amount of work brought from highly reputed conferences and journals viz. IEEE, Elsevier, Springer, ACM and 5% other journals, and conferences on the application of intelligent techniques. The review is broadly categorized according to the type of technique applied viz. (1) Neural networks (NNs), (2) fuzzy logic, (3) genetic algorithm, (4) decision tree, (5) case base reasoning and (6) soft computing. This article basically attention on the importance of benchmark datasets were used for training and testing of all intelligence techniques, type of techniques were applied, and which are the best evaluation metrics. After doing tremendous analysis, we observed that the benchmark data sets are COCOMO, NASA, ISBSG, DEHANAIS prominent datasets, and the evaluation metrics are MMRE, PRED is prominent which is based on total count is presented in Table 8 and 9. Furthermore, we found that Neural Networks technique was used recurrently by the authors and at the same time the techniques applied non regular intervals such as Hybrid, fuzzy logic, decision tree and evolutionary computation. This review is going to be useful for researchers as beginners as and it provides future directions. This would eventually be led to better predict, in the field of Software Cost Estimation.

**Keywords:** Software Cost Estimation (SCE), Neural Networks (NN), Fuzzy Logic(FL), Decision Tree (DT), Genetic Algorithms (GA), Soft Computing (SC).

## INTRODUCTION

In present scenario, modern organizations chose software development as an important activity because of the quality, cost, effort and timeliness of developed software are often crucial forms of an organization to achieve success [1]. As a result of this, there is a proper planning, design, and need more realistic, effective estimation methods before preliminary project development phase. The project estimated prerequisite is vital to achieve the estimated cost, quality analysis, accuracy and to extent maximum profit. To predict the amount of resources and period of a time required to complete a software project is software cost estimation. The duration and resources of the project are estimated on the basis of the size and complexity of the software project in terms of Person-Months (PM). The project size is typically measured as Kilo Lines of Source Code (KLOSC) and Effort Adjustment Factor (EAF), is specified as software complexity. The software complex comprises various cost factors, for instance, the level of the reliability, the programmer skill, the software tools used, and so on. All these factors make sure inherent uncertainty associated with SCE. The methods have been proposed to solve the uncertainty problem, which are two distinct types of effort estimation methods: Algorithmic and Non-Algorithmic. Algorithmic models are widely represented in mathematical sophistication and based on the statistical analysis of historical data (past projects) for example, Software Life Cycle Management, Function Point, Regression models and Constructive Cost Model (COCOMO). The COCOMO model is a regression model hosted by the (Bohem, 1981) best known and the most plausible of all the traditional cost prediction models. The COCOMO model is used to compute effort, time and average staff of the software projects. The Non-Algorithmic methods are based on past experience, and likeness and so on, for instance, Expert Judgment, Price to win, and Analogy. These models were contributed up to some extent to solve vagueness incorporated by cost drivers. All the traditional methods were not succeeding to resolve implicit nonlinearity and intercommunication between the characteristics of the project and effort. The objective of this review paper presents computational intelligence techniques were used to give solutions to the problems which are implicit nonlinearity associated with SCE and intercommunication between the characteristics of the project and effort. In addition, this review makes attention on how important datasets were employed to evaluate the performance of an intelligence estimation method used in the study of SCE. In this process, we discovered that four prominent datasets were recurrently used in order to predict the estimated cost, quality, and improve accuracy of product. Moreover, we have provided most of the evaluation methods which are not presented in any article till now. In fact, we found that two are prominent evaluation methods regularly used.The rest of the paper is organized in the following manner. In section 2, overview of computational intelligence techniques and merits, demerits are presented. 3. Review Evaluation Criterion methods are presented. In the section 4, Review Methodology was presented. In section 5 discussions. In section 6 conclusion of the paper.

## REVIEW OF EMPLOYED TECHNIQUES

This study describes the following intelligent techniques

- Neural Networks (NN)
- Fuzzy Logic (FL)
- Genetic Algorithms (GA)
- Decision Tree (DT)
- Case-Based Reasoning (CBR)
- Soft Computing (Hybrid Intelligent Systems)

### Neural Networks (NN)

Neural Networks [3], which is a computational representation inspired by basic model of the natural nervous system, are highly interconnected neurons that have the skill to learn and thereby secure knowledge and make it accessible for use. They are an enormously parallel distributing processing structure. The main intent of the neural network study is to design a computational machine, to mimic how the human brain to accomplish various computational tasks faster than out-dated systems, are known as Artificial Neural Networks. An artificial neural networks do different tasks, for instance data classification, approximation, pattern-matching, and data clustering. Hence, we needed high speed digital devices to implement artificial neural networks, which make the simulation of the neural network processes feasible. The neurons are connected to each other to propagate the input signal and compute a weighted summation of its input signals and generate net output signals, if the net output signals exceed a certain activation function, then the desired output signals become an excitatory or inhibitory input signals to other processing elements are called neurons, and this process continues up to one or more actual output signals are generated. An Artificial Neural Networks typically used various learning mechanisms to train networks are supervised for example Back propagation, Stochastic, Least Mean Square, unsupervised are Competitive, Hebbian, and Reinforced (Outcome based). An author [4] referred, Artificial Neural Network architectures categorized into diverse types based on their network learning approaches viz. 1) Multilayer Perceptron (MLP), 2) Learning vector Quantization (LVQ), 3) Radial Basis Function Network (RBFN), 4) Probabilistic Neural Network (PNN), 5) Self-Organizing feature Map (SOM), and 6) Counter Propagation Network (CPN). An Artificial Neural Networks can have great mapping ability map input patterns to it is related output patterns as well as can predict new outcome patterns from past trends. The Neural Networks are robust and fault tolerant, therefore they might be 85-90 % accurate.

### Fuzzy Logic (FL)

Fuzzy logic first recommended [6] in 1995 and it was applied to several areas, from control mechanism theory to an artificial intelligence. The Fuzzy logic enlightens a method of mathematics that is employed in classical the inference structure that enables right human intellectual capabilities. The Fuzzy logic element may have a truth value that ranges in degree between 0 and 1. For instance, the outcome of a comparison between two objects could not be a “high” or “low” but 0.38 of tallness. Enlightened [6] that “As complexity growths, precise statements drop meaning and meaningful declarations lose precision”. It provides a right method to handle imprecision that arises in the measurement process. It can fit in both quantitative and qualitative information within a single model. Furthermore, when linguistic elements are used, these degrees may be accomplished by specific meaning viz. (1) ‘many’, (2) ‘low’, (3) ‘medium’, (4) ‘often ‘and (5) ‘few’. The Fuzzy set theory elements have a degree of membership. In case of a classical set theory, the elements in a set are related to membership is measured in binary terms, according to bivalent condition, i.e. describes crisp actions that either do or do not occur. The most widely held Fuzzy Logic Schemes are categorized into three types viz. (1) Pure Fuzzy Logic Scheme, (2) Takagi and Sugeno Fuzzy Scheme, and (3) Fuzzy Logic Scheme. All these schemes have the Fuzzifier and Defuzzifier. The Fuzzifier is the process of maps crisp inputs theory into fuzzy set theory and the Defuzzifier is the reverse process of the Fiuzzifier.

### Decision Tree (DT)

Decision tree proposed by [7] is logically a binary tree that shows how a dependent (target) variable is predicted using the set of independent variables according to the regression model. A Decision tree comprises nodes, where the root node is topmost node and represents all the tuples in a dataset. Further, each internal node split into two nodes using splitting variable represents a test on an attribute using different impurity measures and distance based measures, every branch denotes an outcome of the test, and nodes do not have internal nodes are terminals or leaf nodes. Finally, all the leaf nodes hold target value and this process is known as recursive partitioning. Decision Tree widely used to illustrate classification models, because of intelligible nature that resembles human intellectual. There are dissimilar types of Decision Tree viz. (1) ID3 (Iterative Dichotomiser), (2) C4.5 (a successor of ID3), and (3) classification and Regression Trees (CART). All these techniques adopt a greedy approach in which decision trees are built in a top-down recursive partitioning strategy.

**Genetic Algorithms (GA)**

Genetic Algorithm proposed by [8] is unconventional search and optimization procedure, which imitate some of the natural systems required for evolution, particularly those that trail the principles of Charles Darwin, "survival of the fittest". GA does not need any logic, expertise and prior knowledge to solve particular problem. Hence, GA is employed to solve complex optimization problems, for instance, scheduling a job shop, playing games and organizing the time table. The procedure of a GA started with an arbitrarily selected population of chromosomes represent a problem to be resolved. The fitness function is used to compute the "goodness" of every chromosome. During an evaluation, genetic operators are applied to simulate natural reproduction and mutation of classes.

- Some selection mechanisms for reproduction (i.e. Roulette wheel, tournament, rank, etc.).
- The crossover and mutation operators are used to generate offspring. The amount of possibility of crossover and mutation is nominated based on the application.
- Finally, compute the new generation.

This procedure will terminate either when the optimal criterion is met or the maximum number of generations is reached.

**Case Based Reasoning (CBR)**

Case-Based Reasoning [9] is an advanced methodology for AI and ES. The CBR origins were replicated by aspiration to know how the people remember information, and afterward it was recognized that how the people typically solve problems by recollect how they solved similar kind of problems earlier. Conceptually CBR is typically described by the four activities. (1) Retrieve problem definition from the similar kind of cases. (2) Reuse typically a solution is recommended by a similar kind of case. (3) If necessary, Revise or adapt that solution to well fit the new problem. (4) Hold the new solution once it is typically confirmed or validated. In this review article, we can describe how to apply typical CBR activities to software cost estimation process. 1. Collect all the software cost driver factors from the different sources and applying gray relation grade carefully predicts the indicators which cost factor influenced the effort of software. When the relationship between the characteristics and the effort greater than a certain threshold, this characteristic will be considered as one of the factors, and similarly continue same procedure until we found all the driver factors. Furthermore, we can reuse it to build the cost estimation model. 2. The Aggregate similarity of the case dataset and define the similarity about each of the cost driver factors between the various cases and compute it, thus the final similarity summary is obtained. 3. To estimate the software product cost upon the data of the similarity summary and the actual effort of each case after weighting, we can get the typical undetermined software cost.

**Soft Computing (SC)**

The Soft Computing concept was introduced by [10] is a consortium of integration of various standalone techniques, and several intelligent tools viz. (1) Neural Networks, (2) Genetic Algorithms, (3) Fuzzy Logic, (4) Machine Learning (CBR and decision trees subsumed), (5) Rough Set Theory, and (6) Probabilistic Reasoning in a number of combinations to permutations and exploit typically their strengths. The Hybrid approaches are very balanced and cooperative, rather than competitive. Hence, they have one or more flexible form of data processing skill for control real life vague situations. The main objective of the soft computing exploits is the tolerance for vagueness information handling, approximate reasoning, and incomplete truth in order to accomplish better tractability, better robustness and very low-cost solutions. These are the architectures show the behavior of the soft computing

approaches for instance the Neuro-Fuzzy, the Fuzzy-Neural, the Neuron-Genetic, the Genetic-Fuzzy, the Neuro-Fuzzy-Genetic and Rough-Neural. Moreover, the

Multiclassification systems or an ensemble classifier is treated as soft computing system.

**Table 1:** Merits and demerits of various intelligent techniques [4]

| Technique | Advantages | Disadvantages |
|---|---|---|
| Neural Networks | Robust to errors free in training data, good at classification, prediction, Function approximation, the clustering and an optimization tasks on the subject to neural network architecture | Slow learning rate and convergence process rate are low, an error surface consists multiple local minima and the problem of over fitting. |
| Fuzzy Logic | It is good at human comprehensive fuzzy logic 'if-then' rules; It has not very high computational requirements | The Arbitrary choice of the membership indications skews the results, although the triangular shape most often used one. Secondly, the plethora of choices for membership shapes, connectives for fuzzy sets and defusification operators are the disadvantages |
| Genetic Algorithms | It is good at finding the global optimum of an extremely non-linear, and non-convex without getting trapped in local minima | May not yield a global optimum solution always unless it is augmented by a suitable direct search method |
| Case Based Reasoning | Good at not a big data sets when data appear as cases; it is similar to the human like a decision making | Do not apply to not a small data sets; it is poor in generalization |
| Soft Computing | It strengthens the advantages of intelligent methods while at the same time nullifying their disadvantages | Apparently, it has no shortcomings. However, it does need a good amount of data records, which is not exactly drawbacks nowadays |

## REVIEW OF EMPLOYED EVALUATION METRICS

The evaluation metrics are essential in order to evaluate the performance of the proposed models. Unfortunate, A key query of any estimation technique is whether the predictions are accurate or not; the definition of accuracy is the difference between actual effort, and predicted effort, should be as small as possible. In this article, we can give the metrics which are used to evaluate the accuracy percentage of employed or proposed techniques, which are not presented in the case of any article till now.

The Magnitude of Relative Error (MRE) for each observation i can be found at [11]

$$MRE = \frac{E_i - \hat{E}_i}{E_i} \qquad (1)$$

Where $E_i$ is an Actual effort, $\hat{E}_i$ is Estimated Effort.

The Mean Magnitude of Relative Error (MMRE) for each observation i can be obtained at [9]

$$MMRE = \frac{100}{n}\sum_{i}^{n}\frac{E_i - \hat{E}_i}{E_i} \qquad (2)$$

The MdMRE is the median of all MRE's (Zadeh, 1994).

$$MdMRE = 100 * median(MRE_1, MRE_2, MRE_3, ........MRE_n) \qquad (3)$$

A complementary accuracy measure is $PRED_L$, the fraction of observations for which the predicted effort $\hat{E}_i$ falls within an L percent of actual effort $E_i$ [9]

$$PRED_L = \frac{100}{N}\sum_{i}^{N}\begin{cases}1, & \text{if } MRE \le \frac{L}{100}\\ 0 & \text{otherwise}\end{cases} \qquad (4)$$

Variance Accounted For (VAF): [23]

$$\%VAF = \left[1 - \frac{VAR(E_i - \hat{E}_i)}{VAR(\hat{E}_i)}\right]*100 \qquad (5)$$

Variance Absolute Relative Error (VARE): [23]

$$\%VARE = VAR\left[\frac{ABS(E_i - \hat{E}_i)}{E_i}\right]*100 \qquad (6)$$

Mean Balanced Relative Error (MBRE): [97]

$$ASRE = \frac{1}{|M|}\sqrt{\sum_{i=1}^{|M|}\left(\hat{E}_i - EA_i\right)} \quad (7)$$

Mean Inverted Balanced Relative Error (MIBRE): [97]

$$MIBRE_i = \frac{\left|(\hat{E}_i - E_i)\right|}{\max(\hat{E}_i, E_i)} \quad (8)$$

Average Absolute Error (AAE): [22]

$$AAE = \sum_{i=1}^{n}\frac{(\hat{E}_i - E_i)}{(E_i)} \Big/ n * 100 \quad (9)$$

Mean Absolute Percentage Error (MAPE): ([15]

$$MAPE = \frac{100}{n} * \sum_{i=1}^{n}\left[\frac{(\hat{E}_i - E_i)}{(E_i)}\right] \quad 10)$$

Root Mean Square Error (RMSE): [67]

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(E_i - \hat{E}_i\right)^2} \quad (11)$$

Normalized Root Mean Square Error (NRMSE): [79]

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(E_i - \hat{E}_i\right)^2}{\sum_{i=1}^{n}(E_i)^2}} \quad (12)$$

Relative Error (RE): [23]

$$RE = \frac{ABS\left(E_i - \hat{E}_i\right)}{E_i} \quad (13)$$

Mean Absolute Relative Error (MARE): [23]

$$MARE = Mean(RE) * 100 \quad (14)$$

Normalized mean square error (NMSE): [77]

$$NMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(E_i - \hat{E}_i\right)^2}{\sum_{i=1}^{n}(E_i - E)^2}} \quad (15)$$

Where E is the mean value

Magnitude of Relative Error relative of the Estimate (EMRE): [59]

$$EMRE = \frac{\left(E_i - \hat{E}_i\right)}{\hat{E}_i} \quad (16)$$

Magnitude of Relative Error relative of the Estimate (MEMRE): [59]

MEMRE: Mean (EMRE) (17)

Average square root error (ASRE): [75]

$$ASRE = \frac{1}{|M|}\sqrt{\sum_{i=1}^{|M|}\left(\hat{E}_i - EA_i\right)} \quad (18)$$

Where the dataset contains the total number of modules are represented by |M|, $\hat{E}_i$ indicates the predicted effort expended for the module $M_i$ and $EA_i$ indicates the actual effort expended for module $M_i$, as reported by programmers through the time accounting system.

The formula for computing the accuracy is as follows:

$$Accuracy = \frac{TP + TN}{(TP + TN) + (FP + FN)} \quad (19)$$

Where, TP - True Positive, TN – True Negative, FP - False Positive, and FN - False Negative

## REVIEW METHODOLOGY

We did an analysis on different articles which were published in reputed conferences and journals during 2000-2017. Where we observed the importance of a dataset that was used to train and test the predictable techniques which are used to predict the estimated cost, quality of software and the type of an estimated techniques, how to evaluate accuracy of their proposed techniques. We basically take into account of five attributes which are presented in the following table data set, type of techniques, evaluation metrics, year and reference presented in the Table two to seven.

Table 2 presents the distribution of various datasets are used for training and testing, the kind of evaluation techniques were applied, the diverse form of neural networks viz. Feed Forwarded, Radial Basis Function Network, and Counter Propagation Network were used to predict the accuracy of effort, time and cost of software application. Further, it was found that Multi-Layer-Perceptron with back propagation algorithm being forefront data driven technique in a huge number of Applications.

From Table 3, we detected two Fuzzy techniques (1) Fuzzy Logic and (2) Fuzzy Model. The Fuzzy Logic which is articulated if-then rules and the Fuzzy Model which is implemented fuzzification, and defuzzification. These methods used for a specified data in the form linguistic way of representation to remove uncertainty in input data. Further, it was found that two techniques were being forefront Knowledge- driven technique in a huge number of Applications.

Table 4 described is that all authors employed MMRE, and PRED (0.25) evaluation techniques to assess the performance of proposed methodologies. Moreover, we found that the CART, SVM, MLP, and OLS were equally contributed in their relevance.

From Table 5 it was found that the sheta [59] & [61] has been published two articles. He has proposed a new model of structure articulated to implement operations in a case of mutation, crossover and section to estimate software effort for given the projects sponsored by NASA, Desharnais. The rest of the authors were published one article, respectively.

From Table 6, we observed that pahariya et al. [72], [76], used various stand-alone RBF, DFNFIS SVR, MARS, TREENET, MLR, CART, MLP, CPNN, GP and proposed six hybrid methods are the RGP, GP-RGP, GP-GP, GMDH-GP, and GP-GMDH. Throughout the study 10-fold cross validation is performed. We tested on host techniques on ISBSG dataset. The proposed hybrid methods out formed to all other hybrid methods.

From Table 7, find that distinct technique employed on various datasets for data analysis to derive efficient proposed methods. We have different concrete evaluation methods were applied to assess the performance of the proposed techniques for better effort, minimize cost and complete development within a time. Hence, we found that CBR, Regression, UCP were contributed to work carried out behalf other methods. Finally, an Abbreviation of all the intelligent techniques is given in Appendix A.

## DISCUSSION

We construct the Table 8 from the different tables such as 2,3,4,5,6, and 7 based on the count of data sets were used for training and testing, the number of intelligence techniques was applied to estimate the effort (resources, time and budget) and so on. It was noticed that COCOMO dataset was occupied first place based on total counts (No.of times used in all techniques) is 41 (see Table 8), it was used for analysis and tested across all the techniques presented during the year 2000 to 2017 but unfortunately did not be employed by the evolutionary techniques. The NASA dataset was engaged second and ISBSG third place, total count is 28 and 18 respectively. The both data sets were used for analysis and tested across all the techniques and it was found that the ISBSG dataset used for training and testing except Fuzzy Logic. Further, DESHARNAIS dataset probably got fourth place the total count is 12. It was used for training and tested across all the techniques. We did not take into account if the total count below 11. Therefore, we found that the most prominent datasets used in the study of SCE ware COCOMO, followed by NASA, ISBSG and DESHARNAIS. The PROJECTDATA is artificial dataset, it is not prominent compared to the first four mentioned data sets, but it is still better than the remaining datasets that are KEMETER, IBM, and MAXWELL and so on.

Furthermore, from Table 8 the computational techniques that have been used for predicting estimated effort, time and the budget. It was found that the neural network technique was used recurrently when compared to the other models which is based on the total count is 45 (see Table 8) followed by the hybrid techniques the total count is 31, total count of the fuzzy logic is 10, Finally, the decision tree and evolution computation sharing same total count is 10 and 12 respectively.

We, presented distinguish types of evaluation criterion methods which are used to evaluate the performance of the proposed techniques. From the Table 9, we observed that MMRE and PRED methods were most often used to test against the proposed prediction methods for accuracy of a quality of product based on the total count is 52 and 44 respectively. Moreover, it was noticed that researchers, practitioners shown interest to employ RSME to evaluate performance of proposed prediction techniques for SCE. Hence, MMRE and PRED are prominent performance evaluation methods in the field of study, MRE and MdMRE methods also countable used for evaluating performance of the proposed techniques.

## CONCLUSION

Predict the estimated cost at early stages of development life cycle is a challenging task for the effective management of any software industry. This review basically attention on the importance of the datasets was employed for analysis, types of intelligence and machine learning techniques were applied to predict estimated cost and finally, performance evaluated of prediction methods. From our review, we found that the

COCOMO dataset is the most prominent dataset, followed by NASA, ISBSG and DESHARNAIS dataset. The MMRE and PRED are prominent performance evaluation methods in the field of study. Further, we found that the NEURAL NETWORKS technique was recurrently used when compared to the other models followed by the HYBRID techniques, then FUZZY LOGIC, DECISION TREE and EVOLUTION COMPUTAION in that order. This review is helping to great for research beginners in the field of software cost Estimation.

Furthermore, we suggest the various other machine learning and intelligent techniques can be employed on benchmark data sets to test the quality and accuracy of the software cost estimation as follows:

- Extreme Learning Machine
- Spiking Neural Networks
- Kernel based SVM
- Ant Colony Optimization Techniques Particle Swarm Optimization Techniques etc.
- Ensemble method of all these techniques along with employed techniques

**Table 2:** Description of various Neural Network, Evaluation methods and datasets

| Year | Intelligent Methods | Datasets | Evaluation Methods | References |
|---|---|---|---|---|
| 2001 | MLP | KEPC, SED | MMRE | [5] |
| 2002 | MLP | COMOMO 81, KEMERERAND (15), IBM DPS (24) | MMRE, PRED (0.25) | [13] |
| 2004 | RBFN | ISBSG, COCOMO 81 | MMRE | [14] |
| 2005 | Adaptive NN | COCOMO 81 | MRE | [15] |
| 2006 | RBFN with C-Means or APC-III | COCOMO 81 (63) | MMRE, PRED (0.25) | [16] |
| 2007 | MLP and Regression model | ISBSG, COCOMO (63) | MMRE | [17] |
| | MLP | PROJECT DATA (PROM) | MRE | [18] |
| 2008 | MLP | KEMERER & IBM, COCOMO | MAPE | [19] |
| 2009 | MLP | ALBRECHT, DESHARNAIS, MAXWELL, ISBSG | MMRE, PRED (0.25) | [ 20] |
| 2010 | FFNN | 132 SHORT-SCALE PROJECTS | MMRE | [21] |
| | ANN | COCOMO81, IBMDPS, CF,DESHARNAIS, NASA | MMRE, PRED (0.25) | [22] |
| | FFNN | COCOMO 63 | MMRE, PRED (0.25) | [23] |
| | BPNN | UCPWEIGHTS | AAE | [24] |
| 2011 | RBNN&FFNN | NASA (60) | MMRE, RED | [25] |
| | FFBPNN, CASCADE FFBPNN, ELMAN BPNN | PROJECT DATA | MMRE, BRE, PRED | [26] |
| 2012 | ANN | COCOMO (63) & NASA (93 | MMRE, PRED | [27] |
| | FFNN | COCOMO, NASA2 (93) | MMRE | [28] |
| | DBSCAN-FLANN & UKW-FLANN | COCOMO 81, NASA2, & DESHARNAIS | MMRE, PRED (0.25) | [29] |
| | ANN | 240 PROJECTS | MMRE, PRED | [30] |
| 2013 | ANN | COCOMO 81 (63) | MRE | [31] |
| 2014 | MLP | COCOMO (63) | MMRE | [32] |
| | MLP | NASA (93) | MMRE, BRE | [33] |
| | RBFN | NASA (93) | MMRE, RSME | [34] |
| | ANN | COCOMO 81 | | [35] |
| 2015 | RBFN | COCOMO 81, ISBSG | MMRE | [36] |
| | FFNN | COCOMO 81, NASA | MMRE | [37] |
| 2016 | MLFFANN | COCOMO II | MRE | [109] |

**Table 3:** Description of various Fuzzy Techniques, Evaluation methods and datasets

| Year | Intelligent Methods | Datasets | Evaluation Methods | References |
|---|---|---|---|---|
| 2000 | FL | COCOMO 81 | PRED(20), MRE, SDRE | [38] |
| 2001 | FL | COMOMO 81, | PRED(20) MRE | [39] |
| 2004 | FL | REAL PROJECTS | NOT SPECIFIED | [40] |
| 2005 | FL | COCOMO 81 | MMRE, PRED (20) | [41] |
| 2006 | FLM | PROJECT DATA | | [42] |
| 2009 | FM | ARTIFICIAL DATA | NOT SPECIFIED | [43] [44] |
| 2010 | FL | COCOMO | MMRE, PRED (25) VAF, MMRE, VARE | [45] |
| | FLM | NASA | | |
| 2011 | FLM | COCOMO I (63), NASA (93), PROJECT DATA | MMRE, PRED (25) | [46] |
| | FLS | NASA (93), DESHARNAIS | NOT SPECIFIED | [47] |
| | FL | ARTIFICIAL DATA | MMRE | [48] |
| 2012 | FA, LV | NASA 93, COCOMO 81 | PRED (25) | [49] |
| 2013 | FLM | UCP | MMRE, MdMRE, PRED (0.25), & (0.5) | [50] |
| | FL | WEB DATA SET | MMRE, PRED (0.25) | [51] [52] |
| 2014 | FL | OFF LINE AND ON LINE | MRE | [105] |
| | FLM | NASA | MMRE | [38] |
| 2015 | FL | PROJECT DATA | MMRE | [108] |
| 2016 | FLM | NASA 93 | VAF, MARE, VARE, MBRE, MMRE, PRED (30%) | [110] [112] |
| 2017 | FLF | COCOMONASA | MMRE, PRED(30%, 20%, &10%) | |

**Table 4:** Description of various Machine Learning Techniques, Evaluation methods and datasets

| Year | Intelligent Methods | Datasets | Evaluation Methods | References |
|---|---|---|---|---|
| 2007 | DC, SVM, MLP | NASA (60), USC, SDR | MMRE, PRED (0.25) | [53] |
| 2010 | CART, OLS, NN | ISBSGR11 (5052), COCOMO 81 | MMRE, PRED (0.25), MdMRE | [54] |
| 2012 | CART, OLS, SVM, MLP, ROR, RIR, LMS, MARS, M5, MLP, RBFN, CBR, LS-SVM | ISBSG, MAXWELL, DESHARNAIS | MMRE, PRED (0.25), MdMRE | [55] |
| 2013 | DT, DTF, MLR | DESHARNAIS, ISBSG | MMRE, PRED (0.25), MdMRE | [56] |

**Table 5:** Description of Evolutionary Techniques, Evaluation methods and datasets

| Year | Intelligent Methods | Datasets | Evaluation Methods | References |
|---|---|---|---|---|
| 2001 | GP | DESHARNAIS | MMRE, PRED (0.25) | [57] |
| 2002 | GGGP | ISBSG | MMRE, PRED (0.25), & (50) | [58] |
| 2006 | GA | NASA (18) | MMRE, VAF | [59] |
| 2008 | GA | ISBSG | MMRE | [60] |
| 2010 | GP | NASA (18) | MMRE, VAF | [61] |
| | GP, GGGP | DESHARNAIS | MMRE, MdMRE, PRED (25), MEMRE, MdEMRE | [62] |
| 2011 | GP | 132 SHORT-SCALE PROJECTS | MMRE | [63] |
| 2012 | GA | PROJECT DATA | NOT SPECIFIED | [64] |
| 2013 | GA | ISBSG | NOT SPECIFIED | [65] |
| 2015 | GA | NASA | MMRE | [66] [106] |

| | | | | |
|---|---|---|---|---|
| 2016 | GA | NASA | MMRE,VAF,RMS | |
| 2016 | GA | COCOMO81, COCOMONASA | MRE,ARE,MMRE | [111] |

**Table 6:** Description of Hybrid Techniques, Evaluation methods and datasets

| Year | Intelligent Methods | Datasets | Evaluation Methods | References |
|---|---|---|---|---|
| 2003 | NEURO-FUZZY | COMOMO81, IndustDATA | RE | [67] |
| 2004 | COCOMO & FL | COMOMO (63) | RMSRE, PRED (25) | [68] |
| 2008 | COCOMO & PSO | NASA (18) | RMSE, MMRE | [69] |
| 2009 | FL, & NN | ISBSG | MMRE, PRED (25) & (50) | [70] |
| | FUZZY-GRAY, | COCOMO 81(63) | NOT SPECIFIED | [71] |
| | | | | [72] |
| | RBF, DFNFIS SVR, MARS, TREENET, MLR, CART, MLP, CPNN, GP, GMDH, GP-GP, GP-RGP | ISBSG | RMSE, | |
| 2010 | MR, NN & GA | COCOMO 81(63) | MMRE, PRED (25) | [73] |
| | COCOMO II & NN | NASA (93) | MMRE, PRED (0.25) | [74] |
| | ROUGH SETS & NN | QUANTITATIVE & BUDGET DATA | RE | [75] |
| | RGP, GP-GP, GP-RGP, GMDH-GP, GP-GMDH | ISBSG | RMSE | [76] |
| 2011 | PSO, NN, & COCOMO II SVM & GA | COCOMO 81 (63) | MARE | [77] |
| | NSSVM &GA | COCOMO (62) | MMRE, PRED (0.25) | [78] |
| | | NPD | MAPE, NMSE | [79] |
| 2012 | COCOMO & NN | NASA (93) | MMRE, PRED (0.25) | [80] |
| | COCOMO II & FL | COCOMO 81 (63) | MRE | [81] |
| | FL | DESHARNAIS (81), MAXWELL (62) | MMRE | [82] |
| 2013 | MLP & RBFN | PROJECT DATA SET | MMRE, NRMSE | [83] |
| | FL, NN, ML & COCOMO | NASA | MMRE, PRED (0.25) | [84] |
| 2014 | COCOMO II, HYBRID & ANN | COCOMO 81 (63) | MRE | [85] |
| | SVM & NN | REAL PROJECTS | MMRE | [86] |
| | FL & GA | COCOMO 81 (63) | ACCURACY | [104] |
| | NN &FL | | | |
| 2015 | GA & PSO | NASA | MRE | [87] |
| | COA-CUCKOO | PROJECT DATA | NOT SPEICFIED | [88] |
| 2016 | | NASA6O, 63, & 90, MAXWELL KEMERER, MIYAZAKI | MRE, PRED(N), MAE, MAPE, RMSE, MDMRE, MMRE, MMER | [107] |

**Table 7.** Description of Other Techniques, Evaluation methods and datasets

| Year | Intelligent Methods | Datasets | Evaluation Methods | References |
|---|---|---|---|---|
| 2001 | VPM & COCOMO II | Project's Data | CORRELATION $R^2$ | [89] |
| | COCOMOI | COBOL-Based ISAM | ASRE, MRE | [90] |
| | CCOCOMO | EXPERIENCE PROJECTS (266) | MRE | [91] |
| | ACOCOMO | | | |
| | OLS REGRESSION | | | |
| 2005 | OLS REGRESSION | ISBSG, MAXWELL, | MMRE, PRED (25) | [92] |
| | ORDINAL REGRESSION | COCOMO 81 | | |
| 2006 | OLS REGRESSION | COCOMO 81, NASA, & COCOMO II | MMRE, PRED (30), | [93] |
| 2010 | CBR | COCOMO 81 | NOT SPECIFIED | [94] |
| 2011 | ABC & H-ABC | NASA (63) | NOT SPECIFIED | [95] |

| | | | | |
|---|---|---|---|---|
| | AABM | DESHARNAIS, COCOMO 81, | MMRE, PRED (25), MdMRE | [96] |
| | | ALBRECHT, | | |
| | | MAXWELL, | | |
| | | ISBSG, & CHINA | | [97] |
| | UCP | PROJECT DATA | NOT SPECIFIED | [98] |
| | ABC, SVM, LR, & | NASA (18), NASA (63), | MRE, | |
| | KNN | DESHARNAIS, COCOMO81, | MMRE, MAR, MdMRE, | [99] |
| 2012 | ENSEMBLE, ML, & MRT | KEMERER, | MER, PRED(25), | |
| | | ALBRECHT, | MBRE, MIBRE | |
| | | TELECOM, | | [101] |
| | | MIYAZAKI 94, | | [100] |
| | | MAXWELL, | | |
| | | ISBSG, CHINA | | |
| | | SDR & FINNISH | | [102] |
| | | ISBSG | PRED (25) | [103] |
| | NBC & SWR | | | |
| | | DESHARNAIS, | PRED(25) & (30), MRE | |
| | ANALOGY | COCOMO 81, | | |
| | | KEMERER, | | |
| | | MAXWELL, ALBRECHT, & ISBSG | | |
| | | DESHARNAIS, MIYAZAKI | | |
| | | | MMRE, PRED (0.25), MdMRE | |
| 2103 | ELU-CBR, MAN-CBR, & | PROJECT DATA | | |
| | MIN-CBR | | NOT SPECIFIED | |
| | UCP | | | |

**Table 8:** Data sets and various intelligent techniques

| **Year** | **2000-16** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | **NN** | **FL** | **DT** | **EC** | **HT** | **OT** | **TOTAL COUNT** |
| COCOMO | 17 | 06 | 01 | 01 | 09 | 07 | 41 |
| NASA | 07 | 06 | 01 | 04 | 06 | 04 | 28 |
| ISBSG | 04 | 00 | 03 | 03 | 03 | 05 | 18 |
| COCOMO NASA2 | 01 | 01 | - | - | -01 | - | 03 |
| DESHARNAIS | 02 | 01 | 02 | 02 | 01 | 04 | 12 |
| IBM DSP | 02 | - | - | - | - | - | 02 |
| KEMERER | 01 | - | - | - | 01 | 03 | 05 |
| KEMERERAND | 01 | - | - | - | - | - | 01 |
| MAXWELL | 01 | - | 01 | - | 02 | 04 | 08 |
| ALBRECHT | 01 | - | - | - | - | 03 | 04 |
| PROJECT DATA | 05 | 07 | - | 02 | 05 | 05 | 24 |
| OTHERS | 03 | 01 | 02 | - | 03 | 07 | 16 |

**Table 9:** Different Evaluation methods and various Intelligent Techniques

| Year | 2000-16 | | | | | | |
|---|---|---|---|---|---|---|---|
| **Evaluation Methods** | **NN** | **FL** | **DT** | **EC** | **HT** | **OT** | **TOTAL COUNT** |
| MRE | 4 | 4 | - | 1 | 3 | 4 | 16 |
| MMRE | 17 | 10 | 4 | 7 | 9 | 5 | 52 |
| PRED | 9 | 12 | 4 | 4 | 8 | 7 | 44 |
| MdMRE | - | 1 | 3 | 2 | 1 | 3 | 10 |
| RSME | 1 | - | - | - | 6 | - | 7 |
| NRMSE | - | - | - | - | 1 | - | 1 |
| MBRE | - | 1 | - | 1 | - | 1 | 3 |
| MIBRE | - | - | - | - | - | 1 | 1 |
| MAPE | 1 | - | - | - | 1 | - | 2 |
| MARE | - | 1 | - | - | 1 | - | 2 |
| MEMRE | - | - | - | 1 | 1 | - | 2 |
| ASRE | - | - | - | - | - | 1 | 1 |
| Accuracy | - | - | - | - | 1 | - | 1 |
| VAF | - | 2 | - | 2 | - | - | 4 |
| VARE | - | 2 | - | - | - | - | 2 |
| AAE | 1 | - | - | - | - | | 1 |
| NOT SPECIFIED | - | 3 | - | 2 | - | 4 | 9 |

**Appendix A: Abbreviation and Explanation**

| Intelligent Technique | Description |
|---|---|
| MLP | Multi-Layer-Perceptron |
| RBFN | Radial Basis Function Network |
| FFNN | Feed Forward Neural Network |
| BPNN | Back Propagation Neural Network |
| ANN | Artificial Neural Network |
| FFBPNN | Feed Forward Back Propagation Neural Network |
| DBSCAN | Density Based Spatial Clustering Application of with Noise |
| FLANN | Functional Link Artificial Neural Network |
| UKW | Unsupervised K-Window Clustering |
| FL | Fuzzy Logic |
| FLM | Fuzzy Logic Model |
| FLS | Fuzzy Logic System |
| GP | Genetic Programming |
| GGGP | Grammar Guided Genetic Programming |
| GA | Genetic Algorithms |
| PSO | Particle Swarm Optimization |
| SVR | Support Vector Machine |
| MARS | Multivariate Adaptive Regression Splines |
| DFNFIS | Dynamic Evolving Neuro–Fuzzy Inference System |
| MLR | Multiple Linear Regression |
| CART | Classification and Regression Tree |
| CPNN | Counter Propagation Neural Network |
| GMDH | Group Method of Data Handling |
| MARS | Multivariate adaptive regression splines |
| TREENET | Tree Net |
| RGP | Recurrent Genetic Programming |
| MR | Multiple Regression |
| C-COCOMO | Calibrated COCOMO |
| A-COCOMO | Augmented COCOMO |
| VPM | Vector Prediction Model |
| OLS REGRESSION | Ordinary Least Squares Regression |
| CBR | Case Based Reasoning |
| ABE | Artificial Bee Colony |

| | |
|---|---|
| H-ABE | Halton- Artificial Bee Colony |
| LR | Linear Regression |
| KNN | K-Nearest Neighbors |
| ML | Machine Learning |
| NBC | Naive Bayes Classifier |
| SWR | Stepwise Regression |
| UCP | Use Case Point |
| ELU-CBR | CBR Method with Euclidean distance |
| MAN-CBR | CBR Method with Manhattan distance |
| MIN-CBR | CBR Method with Minkowski distance |

## REFERENCES

[1] Vinaykumar, K., Ravi, V., and Carr, M., "Software cost estimation using soft computing approaches", In Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, pp. 499-518, 2010.

[2] Boehm, B. W., "Software Engineering Economics", Vol. 197, Englewood Cliffs (NJ): Prentice-Hall, 1981.

[3] Praynlin, E., and Latha, P., "Performance analysis of software effort estimation models using neural networks", International Journal of Information Technology and Computer Science (IJITCS), Vol.5, No. 9, pp.101, 2013.

[4] Mohanty, R., Ravi, V., and Patra, M. R., "The application of intelligent and soft-computing techniques to software engineering problems: a review", International Journal of Information and Decision Sciences, Vol.2, No.3, pp.233-272, 2010.

[5] Jun, E. S., and Lee, J. K., "Quasi-optimal case-selective neural network model for software effort estimation", Expert Systems with Applications, Vol.21, No.1, pp.1-14, 2001.

[6] Zadeh, L. A., "Fuzzy sets", Information and control, Vol.8, No.3, pp.338-353,1695.

[7] Gehrke, J., Ganti, V., Ramakrishnan, R., and Loh, W. Y. BOAT, "optimistic decision tree construction", In ACM SIGMOD Record Vol. 28, No. 2, pp. 169-180, 1999.

[8] Holland, J. H., "Adaptation in natural and artificial systems: an introductory analysis with applications to biology", control and artificial intelligence, MIT press, 1992

[9] Kolodner, J. L., "An introduction to case-based reasoning", Artificial intelligence review, Vol.6, No.1, pp.3-34, 1992.

[10] Zadeh, L. A., "Soft computing and fuzzy logic", IEEE software, Vol.11, No.6, pp.48-56, 1994.

[11] Conte. S., Dunsmore, H. and Shen, V., "Software Engineering Metrics and Models", Benjamin / Cummings, Menlo Park. Calif, 1986.

[12] Jørgensen, M. , "An empirical study of software maintenance tasks", Journal of Software: Evolution and Process, Vol.7, No.1, pp. 27-48,1995.

[13] Idri, A., Khoshgoftaar, T. M., and Abran, A. "Can neural networks be easily interpreted in software cost estimation", Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, Vol. 2, pp. 1162-1167, 2002

[14] Idri, A., Mbarki, S., and Abran, A., "Validating and understanding software cost estimation models based on neural networks", IEEE International Conference on Information and Communication Technologies: From Theory to Applications, pp. 433-434, 2004.

[15] Tadayon, N., "Neural network approach for software cost estimation", IEEE International Conference on Information Technology: Coding and Computing, Vol. 2, pp. 815-818, 2005.

[16] Idri, A., Abran, A., and Mbarki, S., "An experiment on the design of radial basis function neural networks for software cost estimation" IEEE International Conference on Information and Communication Technologies, Vol. 1, pp. 1612-1617,2006

[17] deBarcelos Tronto, I. F., da Silva, J. D. S., and Sant'Anna, N., " Comparison of artificial neural network and regression models in Software Effort Estimation", IEEE International Joint Conference on Neural Networks ( IJCNN), pp. 771-776,2007.

[18] Abrahamsson, P., Moser, R., Pedrycz, W., Sillitti, A., and Succi, G., "Effort prediction in iterative software development processes--Incremental versus global prediction models", IEEE First International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 344-353, 2007.

[19] El-Sebakhy, E. A., "New computational intelligence paradigm for estimating the software project effort", IEEE 22nd International Conference on Advanced Information Networking and Applications-Workshops (AINAW 2008), pp. 621-627, 2008.

[20] Li, Y. F., Xie, M., and Goh, T. N., “A study of the non-linear adjustment for analogy based software cost estimation”, Empirical Software Engineering, Vol.14, No.6, pp. 603-643, 2009.

[21] Kalichanin-Balich, I., and Lopez-Martin, C., “ Applying a Feedforward neural network for predicting software development effort of short-scale projects”, IEEE Eighth ACIS International Conference on Software Engineering Research, Management and Applications (SERA), pp. 269-275, 2010.

[22] Jodpimai, P., Sophatsathit, P., and Lursinsap, C., “Estimating software effort with minimum features using neural functional approximation”, IEEE International Conference on Computational Science and Its Applications (ICCSA) pp. 266-273, 2010.

[23] Attarzadeh, I., and Ow, S. H., “Proposing a new software cost estimation model based on artificial neural networks”, IEEE 2nd International Conference on Computer Engineering and Technology (ICCET), Vol. 3, pp. 483-487, 2010.

[24] Ajitha, S., Kumar, T. S., Geetha, D. E., and Kanth, K. R., “Neural network model for software size estimation using a use case point approach”, IEEE International Conference on Industrial and Information Systems (ICIIS), pp. 372-376, 2010.

[25] Dave, V. S., and Dutta, K., “Neural network based software effort estimation & evaluation criterion MMRE”, IEEE 2nd International Conference on Computer and Communication Technology (ICCCT), pp. 347-351, 2011.

[26] Ghose, M. K., Bhatnagar, R., and Bhattacharjee, V., “Comparing some neural network models for software development effort prediction”, IEEE 2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS), pp.1-4, 2011.

[27] Kaushik, A., Soni, A. K., and Soni, R., “An adaptive learning approach to software cost estimation”, National IEEE Conference on Computing and Communication Systems (NCCCS), pp.1-6, 2012.

[28] Attarzadeh, I., Mehranzadeh, A., and Barati, A., “Proposing an enhanced artificial neural network prediction model to improve the accuracy in software effort estimation’, IEEE Fourth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), pp.167-172, 2012.

[29] Benala, T. R., Dehuri, S., Mall, R., and ChinnaBabu, K., “Software effort prediction using unsupervised learning (clustering) and functional link artificial neural networks”, IEEE World Congress on Information and Communication Technologies (WICT), pp.115-120, 2012.

[30] Nassif, A. B., Capretz, L. F., and Ho, D., “Software effort estimation in the early stages of the software life cycle using a cascade correlation neural network model”, IEEE 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), pp.589-594, 2012.

[31] Saraç, Ö. F., and Duru, N., “A novel method for software effort estimation: Estimating with boundaries”, IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp.1-5, 2013.

[32] Madheswaran, M., and Sivakumar, D., “Enhancement of prediction accuracy in COCOMO model for software project using neural network”, IEEE International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp.1-5, 2014

[33] Praynlin, E., and Latha, P., “Software Effort Estimation Models Using Radial Basis Function Network”, World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol. 8, No.1, pp. 258-263, 2014.

[34] Kumar, G., and Bhatia, P. K., “Automation of software cost estimation using neural network technique”, International Journal of Computer Applications, Vol. 98, No. 20, 2014.

[35] Mukherjee, S., and Malu, R. K., “ Optimization of the project effort estimate using neural network”, IEEE International Conference on Advanced Communication, Control and Computing Technologies (ICACCCT), pp. 406-410, 2014.

[36] Idri, A., Hassani, A., and Abran, A., “RBFN networks-based models for estimating software development effort: A cross-validation study”, IEEE Symposium Series on Computational Intelligence, pp. 976-983, 2015.

[37] Sarno, R., and Sidabutar, J., “Comparison of different Neural Network architectures for software cost estimation”, IEEE International Conference on, Computer, Control, Informatics and its Applications (IC3INA), pp. 68-73, 2015.

[38] Idri, A., Abran, A., and Kjiri, L., “ COCOMO cost model using fuzzy logic”, 7th International Conference on Fuzzy Theory & Techniques, Vol. 27,

2000.

[39] Idri, A., and Abran, A., "A fuzzy logic based set of measures for software project similarity: validation and possible improvements", IEEE Proceedings of Seventh International Software Metrics Symposium, pp. 85-96, 2001.

[40] Braz, M. R., and Vergilio, S. R., "Using fuzzy theory for effort estimation of object-oriented software", 16th IEEE International Conference on Tools with Artificial Intelligence ( ICTAI), pp.196-201, 2004

[41] Martin, C. L., Pasquier, J. L., Yanez, C. M., and Tornes, A. G., "Software development effort estimation using fuzzy logic: a case study", IEEE Sixth Mexican International Conference on Computer Science, pp. 113-120, 2005.

[42] Lopez-Martin, C., Yanez-Marquez, C., and Gutierrez-Tornes, A.,"A fuzzy logic model based upon reused and new & changed code for software development effort estimation at personal level", IEEE 15th International Conference on Computing, pp. 298-303, 2016.

[43] Attarzadeh, I., and Ow, S. H., "Proposing a new software cost estimation model based on artificial neural networks", IEEE 2nd International Conference on Computer Engineering and Technology (ICCET), Vol. 3, pp. 483-487, 2010.

[44] Verma, H. K., and Sharma, V., "Handling imprecision in inputs using fuzzy logic to predict effort in software development", IEEE 2nd International Advance Computing Conference (IACC), pp. 436-442, 2010.

[45] Hari, C. V., PVGD, P. R., Jagadeesh, M., & Ganesh, G. S., "Interval Type-2 Fuzzy Logic for Software Cost Estimation Using TSFC with Mean and Standard Deviation", International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), pp. 40-44, 2010

[46] Attarzadeh, I., and Ow, S. H., "Improving the estimation accuracy of the COCOMO II using an adaptive fuzzy logic model", IEEE International Conference on Fuzzy Systems, pp. 2458-2464, 2011.

[47] Malathi, S., and Sridhar, S., "A Classical Fuzzy Approach for Software Effort Estimation on Machine Learning Technique", arXiv preprint arXiv: 1112.3877, 2011.

[48] Sadiq, M., Mariyam, F., Alil, A., Khan, S., and Tripathi, P., "Prediction of software project effort using fuzzy logic", IEEE 3rd International Conference on Electronics Computer Technology (ICECT), Vol. 4, pp. 353-358, 2011.

[49] Malathi, S., and Sridhar, S., "Optimization of Fuzzy Analog for Software Cost Estimation using Linguistic Variables", International Conference on Modeling, Optimization, and Computing, Procidia Engineering, pp. 1-14, 2012.

[50] Azzeh, M., and Nassif, A. B., "Fuzzy Model Tree for Early Effort Estimation", IEEE 12th International Conference on Machine Learning and Applications (ICMLA), Vol. 2, pp. 117-121, 2013.

[51] Idri, A., and Zahi, A., "Software cost estimation by classical and Fuzzy Analogy for Web Hypermedia Applications: A replicated study", In Computational Intelligence and Data Mining (CIDM), pp. 207-213, 2013.

[52] Patil, L. V., Shivale, N. M., Joshi, S. D., & Khanna, V., " Improving the accuracy of CBSD effort estimation using fuzzy logic", IEEE International Symposium on Advance Computing Conference (IACC), pp. 1385-1391, 2014.

[53] Baskeles, B., Turhan, B., and Bener, A., "Software effort estimation using machine learning methods", IEEE 22nd international symposium on Computer and information sciences, Iscis , pp. 1-6, 2007.

[54] Setiono, R., Dejaeger, K., Verbeke, W., Martens, D., and Baesens, B., " Software effort prediction using Regression rule extraction from neural networks", 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Vol. 2, pp. 45-52, 2010.

[55] Dejaeger, K., Verbeke, W., Martens, D., and Baesens, B., "Data mining techniques for software effort estimation: a comparative study", IEEE transactions on Software Engineering, Vol. 38, No.2, pp. 375-397, 2012.

[56] Nassif, A. B., Azzeh, M., Capretz, L. F., and Ho, D., "A comparison between decision trees and decision tree forest models for software development effort estimation", IEEE Third International Conference on Communications and Information Technology (ICCIT), pp. 220-224, 2013.

[57] Burgess, C. J., and Lefley, M., "Can genetic programming improve software effort estimation?: A comparative evaluation", Information and Software Technology, Vol. 43, No.14, pp. 863-873. 2001.

[58] Shan, Y., McKay, R. I., Lokan, C. J., and Essam, D. L., "Software project effort estimation using genetic programming", IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions, Vol. 2, pp. 1108-1112, 2002.

[59] Sheta, A. F., "Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects", Journal of Computer Science, Vol.2, No.2, pp. 118-123, 2006.

[60] Ahmed, F., Bouktif, S., Serhani, A., and Khalil, I., "Integrating function point project information for improving the accuracy of effort estimation", IEEE the Second International Conference on Advanced Engineering Computing and Applications in Sciences, pp. 193-198, 2008.

[61] Alaa, F. S., and Al-Afeef, A., "A GP effort estimation model utilizing a line of code and methodology for NASA software projects", IEEE 10th International Conference on, Intelligent Systems Design and Applications (ISDA), pp. 290-295, 2010.

[62] Ferrucci, F., Gravino, C., Oliveto, R., and Sarro, F., "Genetic programming for effort estimation: an analysis of the impact of different fitness functions", IEEE Second International Symposium on Search Based Software Engineering (SSBSE), pp. 89-98, 2010.

[63] Chavoya, A., Lopez-Martin, C., and Meda-Campa, M. E., "Applying genetic programming for estimating software development effort of short-scale projects", IEEE Eighth International Conference on Information Technology: New Generations (ITNG), pp.174-179, 2011.

[64] Galinina, A., Burceva, O., and Parshutin, S., "The optimization of COCOMO model coefficients using genetic algorithms", Information Technology and Management Science, Vol.15, No.1, pp. 45-51, 2012.

[65] Manikavelan, D., and Ponnusamy, R. "To find the accurate software cost estimation using Differential Evaluation Algorithm", IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1-4, 2013.

[66] Algabri, M., Saeed, F., Mathkour, H., and Tagoug, N., "Optimization of soft cost estimation using genetic algorithm for NASA software projects", IEEE 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW), pp. 1-4, 2015.

[67] Huang, X., Capretz, L. F., Ren, J., and Ho, D., "A neuro-fuzzy model for software cost estimation", IEEE Third International Conference on Quality Software, pp.126-133, 2003.

[68] Saliu, M. O., Ahmed, M., and AlGhamdi, J., "Towards adaptive soft computing based software effort prediction", IEEE National Annual Meeting on Fuzzy Information Processing Systems (NAFIPS), Vol. 1, pp. 16-21, 2004.

[69] Sheta, A., Rine, D., and Ayesh, A., "Development of software effort and schedule estimation models using soft computing techniques", IEEE Congress on Evolutionary Computation (CEC), pp. 1283-1289, 2008.

[70] Wong, J., Ho, D., and Capretz, L. F., "An investigation of using neuro-fuzzy with software size estimation", IEEE International Conference on Software Engineering Workshop on Software Quality (WOSQ), pp. 51-58, 2009.

[71] Zhang, B., and Zhang, R. , "Evaluation model of software cost estimation methods based on fuzzy-grey theory", IEEE Fourth International Conference on Internet Computing for Science and Engineering (ICICSE), pp. 52-55, 2009.

[72] Pahariya, J. S., Ravi, V., and Carr, M., "Software cost estimation using computational intelligence techniques", IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 849-854, 2009.

[73] Li, Z., "Intelligently Predict Project Effort by Reduced Models Based on Multiple Regressions and Genetic Algorithms with Neural Networks", IEEE International Conference on E-Business and E-Government (ICEE), pp. 1536-1542, 2010.

[74] Attarzadeh, I., and Ow, S. H. "A novel soft computing model to increase the accuracy of software development cost estimation", IEEE 2nd International Conference on Computer and Automation Engineering (ICCAE), Vol. 3, pp. 603-607, 2010.

[75] Wang, X. Z., and Xing, L. Y., "Application of rough set and neural network in engineering cost estimation", IEEE International Conference on Management and Service Science (MASS), pp. 1-4, 2010.

[76] Pahariya, J. S., Ravi, V., Carr, M., and Vasu, M., "Comutational Intelligence Hybrids Applied to Software Cost Estimation", International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM) ISSN: 2150-7988 Vol.2, pp.104-112, 2010.

[77] Hari, C. V., Sethi, T. S., Kaushal, B. S. S., and Sharma, A., "CPN-a hybrid model for software cost estimation", IEEE Conference on Recent Advances in Intelligent Computational Systems (RAICS), pp. 902-906, 2011

[78] Lin, J. C., Chang, C. T., and Huang, S. Y. , "Research on software effort estimation combined

with genetic algorithm and support vector regression", IEEE International Symposium Computer Science and Society (ISCCS), pp. 349-352, 2011.

[79] Mousavi, S. M., and Iranmanesh, S. H., "Least squares support vector machines with genetic algorithm for estimating costs in NPD projects", IEEE 3rd International Conference on Communication Software and Networks (ICCSN), pp. 127-131, 2011.

[80] Singh, B. K., and Misra, A. K., "An alternate soft computing approach for effort estimation by enhancing constructive cost model in evaluation method", International Journal of Innovation, Management and Technology, Vol.3, No.3, pp. 272, 2012.

[81] Kad, S., and Chopra, V. , "Software development effort estimation using soft computing", International Journal of Machine Learning and Computing, Vol. 2, No.5, pp. 548, 2012.

[82] Bardsiri, V. K., Jawawi, D. N. A., Hashim, S. Z. M., and Khatami. E, "Increasing the accuracy of software development effort estimation using projects clustering", IET software, Vol.6, No.6, pp. 461-473, 2012.

[83] Satapathy, S. M., Kumar, M., and Rath, S. K., "Class point approach for software effort estimation using soft computing techniques", IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 178-183, 2013.

[84] Vipul Kumar., and Avadhesh Kumar, "Developing Models for Various Soft Computing Techniques for Software Effort Estimation", Proceeding 4th International Conference, New Delhi, India, ISBN: 978-93-82702-42-9, 2013.

[85] Patil, L. V., Waghmode, R. M., Joshi, S. D., and Khanna, V., "Generic model of software cost estimation: A hybrid approach", IEEE International Advance Computing Conference (IACC), pp. 1379-1384, 2014.

[86] Iwata, K., Liebman, E., Stone, P., Nakashima, T., Anan, Y., and Ishii, N., "Bin-Based Estimation of the Amount of Effort for Embedded Software Development Projects with Support Vector Machines", In Computer and Information Science 2015, Springer International Publishing, pp.157-169, 2016.

[87] Sarno, R., and Sidabutar, J., "Improving the accuracy of COCOMO's effort estimation based on neural networks and fuzzy logic model", IEEE International Conference on Information & Communication Technology and Systems (ICTS), pp. 197-202, 2015.

[88] Wang, X., Jiang, X., and Shi, H., "Prioritization of test scenarios using a hybrid genetic algorithm based on UML activity diagram", 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 854-857, 2015.

[89] Hastings, T. E., and Sajeev, A. S. M., "A vector-based approach to software size measurement and effort estimation", IEEE Transactions on Software Engineering, Vol. 27, No.4, pp. 337-350, 2001.

[90] Smith, R. K., Hale, J. E., and Parrish, A. S., "An empirical study using task assignment patterns to improve the accuracy of software effort estimation", IEEE Transactions on software engineering, Vol. 27, No.3, pp. 264-271, 2001.

[91] Strike, K., El Emam, K., and Madhavji, N., "Software cost estimation with incomplete data", IEEE Transactions on Software Engineering, Vol. 27, No.10, pp. 890-908, 2001.

[92] Sentas, P., Angelis, L., Stamelos, I., and Bleris, G., "Software productivity and effort prediction with ordinal Regression", Information and software technology, Vol. 47, No.1, pp. 17-29, 2005.

[93] Menzies, T., Chen, Z., Hihn, J., and Lum, K., "Selecting best practices for effort estimation", IEEE Transactions on Software Engineering, Vol.32, No.1, 2006.

[94] Jiang, G., Wang, Y., and Liu, H., "Research on software cost evaluation model based on case-based reasoning", IEEE Second World Congress on Software Engineering (WCSE), Vol. 2, pp. 338-341, 2010.

[95] Sharma, T. K., and Pant, M., "Halton based initial distribution in artificial bee colony algorithm and its application in software effort estimation", IEEE Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), pp. 80-84. 2011.

[96] Azzeh, M., "Model tree based adaption strategy for software effort estimation by analogy", IEEE 11th International Conference on, Computer and Information Technology (CIT), pp. 328-335, 2011.

[97] Yavari, Y., Afsharchi, M., and Karami, M., "Software complexity level determination using software effort estimation use case points metrics", IEEE 5th Malaysian Conference Software Engineering (MySEC), pp. 257-262, 2011.

[98] Khalifelu, Z. A., and Gharehchopogh, F. S.,

"Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation", Procedia Technology, Vol.1, pp. 65-71, 2012.

[99] Kocaguneli, E., Menzies, T., and Keung, J. W. "On the value of ensemble effort estimation", IEEE Transactions on Software Engineering, Vol. 38, No. 6, pp. 1403-1416, 2012.

[100] Kocaguneli, E., Menzies, T., Bener, A., and Keung, J. W., "Exploiting the essential assumptions of analogy-based effort estimation", IEEE Transactions on Software Engineering, Vol.38, No.2, pp. 425-438, 2012.

[101] Fernandez-Diego, M., Elmouaden, S., and Torralba-Martinez, J. M., "Software Effort Estimation using NBC and SWR: A comparison based on ISBSG projects", IEEE Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA), pp.132-136, 2012.

[102] Wu, D., Li, J., and Liang, Y. "Linear combination of multiple case-based reasoning with optimized weight for software effort estimation", The Journal of Supercomputing, pp.1-21, 2013.

[103] Azzeh, M., "Software cost estimation based on use case points for global software development', IEEE 5th International Conference on Computer Science and Information Technology (CSIT), pp. 214-218, 2013.

[104] Saraswathi, S., and Kannan, N., "Soft computing based estimation of software development effort", International Conference on Information Communication and Embedded Systems (ICICES), pp.1-6, 2014.

[105] Kushwaha, N., "Software cost estimation using the improved fuzzy logic framework", IEEE Conference on IT in Business, Industry and Government (CSIBIG), pp. 1-5, 2014.

[106] Sachan, R. K., Nigam, A., Singh, A., Singh, S., Choudhary, M., Tiwari, A., & Kushwaha, D. S., "Optimizing Basic COCOMO Model Using Simplified Genetic Algorithm", Procedia Computer Science, 89, 492-498, 2016.

[107] Miandoab, E. E., and Gharehchopogh, F. S., " A novel hybrid algorithm for software cost estimation based on cuckoo optimization and k-nearest neighbors algorithms", Engineering, Technology & Applied Science Research, 6 (3), pp-1018, 2016.

[108] Nosrati Nahook H., **"**The comparison of software cost estimation methods using fuzzy set theory", Scientific Journal of Review, 4(9) 124-132 ISSN 2322-2433, 2015.

[109] Rijwani, P., and Jain, S., "Enhanced software effort estimation using multi layered feed forward artificial neural network technique", Procedia Computer Science, 89, 307-312, 2016.

[110] Sree, P. R., and SNSVSC, R.., "Improving Efficiency of Fuzzy Models for Effort Estimation by Cascading & Clustering Techniques", Procedia Computer Science, *85*, 278-285, 2016.

[111] Kumari, S., and Pushkar, S., " A Framework for Analogy-based Software Cost Estimation using Multi-objective Genetic Algorithm", In Proceedings of the World Congress on Engineering and Computer Science, Vol. 1, 2016.

[112] Jain, R., Khalid Kaleem M., and Bekuma, Y., "Analysis of Software Development Effort Estimation Using Fuzzy Logic Functions with COCOMO II Estimation", International Journal of Scientific & Engineering Research, Volume 8, Issue 1, January-2017.