International Journal of Art & Humanity Science (IJAHS) e-ISSN: 2349-5235,
www.ijahs.com, Volume 4 Issue 2, (Mar-April 2017), PP. 1-9

"SOFTWARE COST ESTIMATION MODEL BASED ON GENETIC ALGORITHMS: OOSD PERSPECTIVES"

Syed Ali Mehdi Zaidi*

Research Scholar, S. V. University,
Gajraula, Amroha, UP., India

Dr. Vinodani Katiyar
Professor, SRM, University,
Lucknow, UP., India,

Prof. (Dr.) S. Qamar Abbas

Director, AIMT,
Lucknow, UP., India†

Mohd. Islam
Research Scholar, S. V. University,
Gajraula, Amroha, UP., India

Abstract:

Software cost estimation is one of important activity of software development. Software cost estimation plays an important role in software engineering practice, often determining the success or failure of contract negotiation and project execution. Cost estimation's deliverables such as effort, schedule, and staff requirements are valuable information for project formation and execution. Genetic Algorithm can offer some significant improvements in accuracy and has the potential to be a valid additional tool for software effort estimation. It is a non-parametric method since it does not make any assumption about the distribution of the data and derives equations according only to the fitted values.

In this study, applicability and capability of Genetic Algorithm techniques for application in software cost estimation as a predictive tool has been investigated. It is seen that GA models are very robust, characterised by fast computation, capable of handling the noisy and approximate data that are typical of data used here for the present study. From the analysis of the results given earlier it is seen that GA has been able to perform well for the prediction of effort estimation. Due to the presence of non-linearity in the data, it is an efficient quantitative tool. The studies has been carried out using MATLAB simulation environment.

Keywords: cost estimation; model based.

Introduction

1.1 Cost Estimation

The estimation of costs related to the software development is one of the major issues in the software industry. The overall process of developing a cost estimation model for software is not different from the process for estimating any other element of cost. There are, however, aspects of the process that are peculiar to software estimating. Some of the unique aspects of software estimating are driven by the nature of software as a product. Other problems are created by the nature of the estimating methodologies. Many of the problems that plague the development effort itself is responsible for the difficulty encountered in estimating that effort. One of the first steps in any estimate is to understand and define the system to be estimated. Software, however, is in tangible, in visible, and intractable.

Software cost estimation is a complex activity that requires knowledge of a number of key attributes about the project for which the estimate is being constructed. Cost estimating is sometimes termed "parametric

* Typeset names in 8 pt Times Roman, uppercase. Use the footnote to indicate the present or permanent address of the author.

† State completely without abbreviations, the affiliation and mailing address, including country. Typeset in 8 pt Times Italic.

estimating" because accuracy demands understanding the relationships among scores of discrete parameters that can affect the outcomes of software projects, both individually and in concert.

The several approaches for the cost estimation techniques are developed. It is classified into following: Model Based -SLIM, checkpoint, SEER, COCOMO, Expertise-Based, Delphi, Rule-Based. Dynamics-Based, Abdel-Hamid Madnick, Learning Oriented Neural, Case-based, Regression-Based, Robust, Composite, Bayesian, COCOCMO-II. Each technique has their own significance and even its disadvantages are also highlighted. This paper concludes that no one model or single method should be favoured over others. The key to achieve the goal i.e. estimation, can be done through variety of tools and methods and then work upon the area that what reasons effects estimation.

1.2 Genetic Algorithms

Genetic algorithms are a type of optimization algorithm, meaning they are used to find the optimal solution to a given computational problem that maximizes or minimizes a particular function. Genetic algorithms represent one branch of the field of study called evolutionary computation, in that they imitate the biological processes of reproduction and natural selection to solve for the `fittest' solutions [1]. Like in evolution, many of a genetic algorithm's processes are random, however this optimization technique allows one to set the level of randomization and the level of control [1]. These algorithms are far more powerful and effcient than random search and exhaustive search algorithms, yet require no extra information about the given problem. This feature allows them to find solutions to problems that other optimization methods cannot handle due to a lack of continuity, derivatives, linearity, or other features.

Genetic Algorithms (GA) are direct, parallel, stochastic method for global search and optimization, which imitates the evolution of the living beings, described by Charles Darwin. GA are part of the group of Evolutionary Algorithms (EA). The evolutionary algorithms use the three main principles of the natural evolution: reproduction, natural selection and diversity of the species, maintained by the differences of each generation with the previous. Genetic Algorithms works with a set of individuals, representing possible solutions of the task. The selection principle is applied by using a criterion, giving an evaluation for the individual with respect to the desired solution. The best-suited individuals create the next generation. The large variety of problems in the engineering sphere, as well as in other fields, requires the usage of algorithms from different type, with different characteristics and settings.

1.3 Object-Oriented Technology

Object-oriented technology, aims to overcome most of the problems associated with the traditional software technologies. Reusability, high modularity, and the innovative approach to design, are expected to increase productivity in the production process. However, the criticality of cost estimation is increased by the change in the technological paradigm. Moreover, the existing techniques were developed according to the traditional software process and languages. The rapid growth of the object-oriented industry and the big capitals committed by many companies' calls for innovative models.

Related Research Works

A number of studies have been published to address cost estimation models and framework for software development and design phase. Existing studies are investigated and their contents and limitations are as follows:

• Lalit V. Patil, et. al., (2014) There are so many models available categorized into algorithmic and non-algorithmic model each of their strengths and weakness. The authors proposed a hybrid approach, which consists of Functional Link Artificial Neural Network (FLANN) and COCOMO-II with training algorithm. FLANN reduces the computational complexity in multilayer neural network. It does not have any hidden layer, and it has fast learning ability.

• Pushpendra K Rajput, Geeta Sikka, and Aarti, (2014), proposed a hybrid model that exploits the uncertainty using clustering the data. In this proposed model they used Genetic Algorithm (GA) combined with COCOMO model on clustered data. Model carries the desirable features of neural network, including learning ability to classify the new project for using the COCOMO model with best fit parameters. The best parameters of COCOMO model can be found for each cluster. The they made comparison of estimated effort with original COCOMO model which can be applied on larger data sets. This scheme also avoids the problem of different estimated cost of similar projects.

• Rahul Chaudhary, et. al., (2013), showed that they can estimate and compare the cost and effort more accurately by using three technologies which are very prominent; they are Grouping Methodologies, Object Oriented Metrics and COCOMO II. All this work helps a manager or Estimator or User of Software to use the previous work (project) in new Real Time Project i.e. there are many references available for continuing to new Real Time Project and secondly, when same goal project is developed by two different logics, then this Tool helps to compare between both Real Time Project in a single Dynamic window on the basic of Object Oriented

Metric. They further compared the software project cost estimation methods based on grouping/groups as new methods that estimates software project cost accurately and are then compared between both the window (or projects result) result and help to fetch out more accurate and correct comparison that helps user/manager to select best old work for their future project.

• K.Ramesh, et. al., (2013), analysed algorithmic modes and non-algorithmic models in the existing models and provided in depth review of software and project estimation techniques existing in industry and literature based on the different test datasets along with their advantages and disadvantages.

• Tharwon Arnuphaptrairong, (2012), analyzed software sizing articles reviewed from the literature and presented the development, and achievements of software size measurement. From the literature review it was found that technologies and techniques related to requirement gathering, and software analysis and design, such as, Structured Analysis and Design Method (SSADM), and Object-oriented Analysis and Design (OOAD), had impacted on the size measurement models. This is because they are directly related to the software functionality. Significant future challenges for software sizing is probably the sizing for new product forms which include requirement or architectural specifications, stories and component-based development. They concluded that besides the new product forms, the new process forms.

• Gurdev Singh, et. al., (2011) studied different type of software metrics which are used during the software development. They showed that a metrics program that is based on the goals of an organization will help communicate, measure progress towards, and eventually attain those goals. People will work to accomplish what they believe to be important. Well-designed metrics with documented objectives can help an organization obtain the information it needs to continue to improve its software products, processes, and services while maintaining a focus on what is important. A practical, systematic, start-to-finish method of selecting, designing, and implementing software metrics is a valuable aid.

Dataset Used for Validation

The dataset (Table-1) from forty Java systems is derived during two successive semesters of graduate courses on Software Engineering. The use of such data in the validation process has provided initial experimental evidence of the effectiveness of the Class Point approach. It is clear that the use of student's projects may threaten the external validity of the experiment and, hence, for the assessment of the method; further analysis is needed by using data coming from the industrial world. Nevertheless, we have worked to make the validation process as accurate as possible. For developing the GA models MATLAB platform will be used.

Table 1: List of Attributes used for model development.

Sr. No.	EFH	NEM	NSR	NOA
1	286	142	97	170
2	396	409	295	292
3	471	821	567	929
4	1016	975	723	755
5	1261	997	764	1145
6	261	225	181	400
7	993	589	944	402
8	552	262	167	260
9	998	697	929	385
10	180	71	218	77
11	482	368	504	559
12	1083	789	362	682
13	205	79	41	98
14	851	542	392	508
15	840	701	635	770
16	1414	885	701	1087

17	279	97	387	65
18	621	382	654	293
19	601	387	845	484
20	680	347	870	304
21	366	343	264	299
22	947	944	421	637
23	485	409	269	451
24	812	531	401	520
25	685	387	297	812
26	638	373	278	788
27	1803	724	1167	1633
28	369	192	126	177
29	439	169	128	181
30	491	323	195	285
31	484	363	398	444
32	481	431	362	389
33	861	692	653	858
34	417	345	245	389
35	268	218	187	448
36	470	250	512	332
37	436	135	121	193
38	428	227	147	212
39	436	213	183	318
40	356	154	83	147

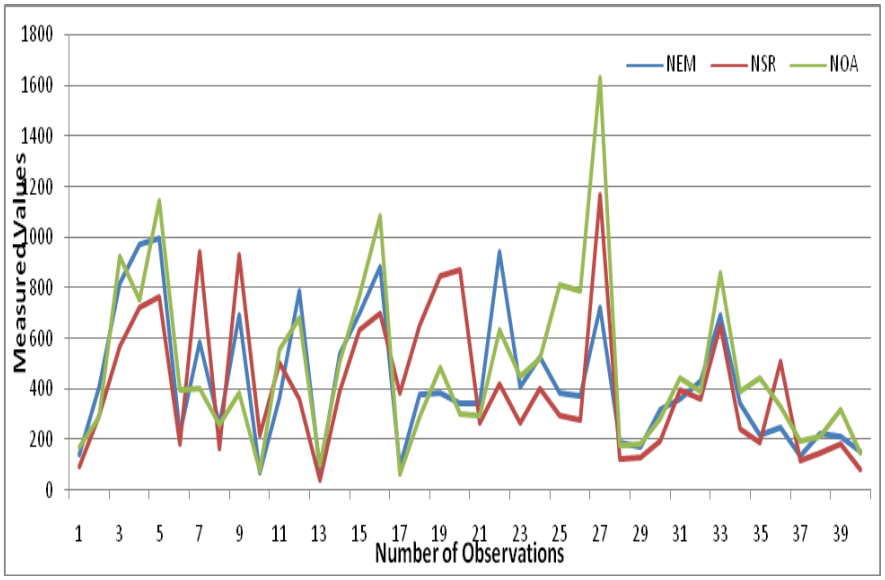
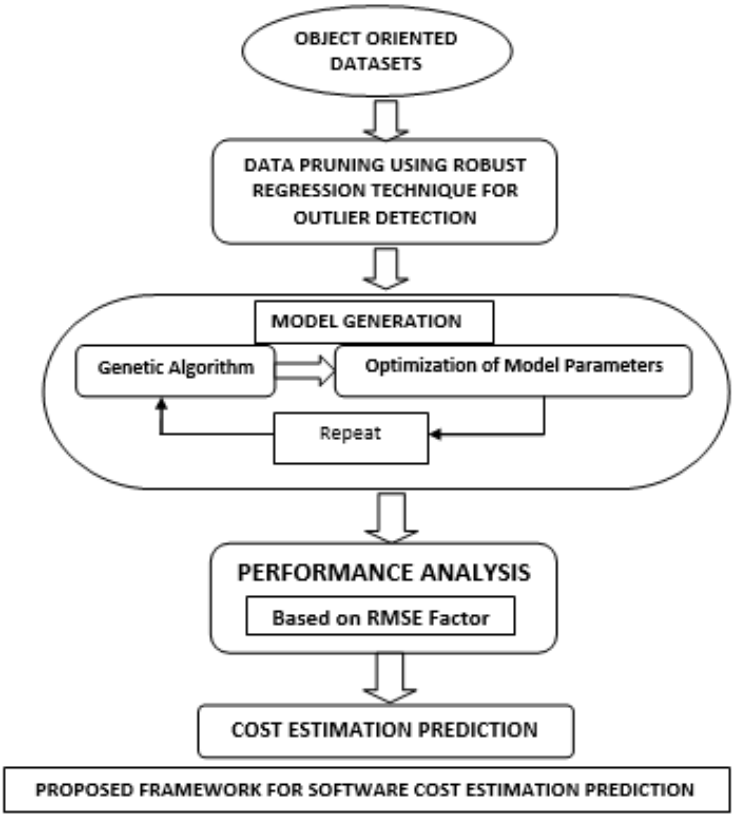


Fig. 1: The plot of the various input and output variables are shown above.

Fig. 2: **Proposed Framework.**



Validation of the Framework

Initially the dataset will be pre-processed, for the detection of the outliers using Robust Linear Regression Technique. This will be done by assigning a weight to each data point. Weighting is done automatically and iteratively using a process called iteratively reweighted least squares. In the first iteration, each point is assigned equal weight and model coefficients are estimated using ordinary least squares. At subsequent iterations, weights are recomputed so that points farther from model predictions in the previous iteration are given lower weight. Model coefficients are then recomputed using weighted least squares. The process continues until the values of the coefficient estimates converge within a specified tolerance.

Next, the model so obtained will be later on subjected to optimization of its model parameters using Genetic Algorithm optimization technique so as to arrive at a better software cost estimation prediction accuracy. The genetic operators such selection, crossover and mutation shall be used. GA runs to generate solutions for successive generations. Hence the quality of the solutions in successive generations improves. The process is terminated when an optimum solution is found.

Finally, the performance of the model shall be analysed based on RMSE and MAE factor. The general framework for the present work is given above fig 2.

4.1 GA Model

Genetic Algorithms (GAs) are heuristic search algorithm based on the ideas of natural selection and genetic. GA is an approach to inductive learning. GA works in an iterative manner. It uses fitness measure to solve the problem. Standard GA uses genetic operators such selection, crossover and mutation. GA runs to generate solutions for successive generations. Hence the quality of the solutions in successive generations improves, the process is terminated when an optimum solution is found. The functions of genetic operators are as follows:

• Selection: Use a fitness function to evaluate the current solution. Where fitness is a comparable measure of how efficiently a chromosome solves the problem at hand.

• Crossover: Crossover develops new elements for the population by combining parts of two elements currently in the population.

• Mutation: Alters the new solutions in the search for better solutions.

In Genetic algorithm the operators are repeatedly applied to a population. This generational process is repeated until a termination condition has been reached.

• Fitness Functions

The fitness function is the function to optimize. For standard optimization algorithms, this is known as the objective function. One has to find the minimum of the fitness function. Write the fitness function as a file or anonymous function, and pass it as a function handle input argument to the main genetic algorithm function.

The fitness function used in the present GA formulation is as below:

Minimize y = abs (a - (α + b*x(1) + c*x(2) + d*x(3))); (1)

Where;

y = minimization value of the objective function,

abs = absolute value,

a = observed value of the effort estimation,

α = constant value obtained from the regression equation,

b = NEM,

c = NSR,

d = NOA,

x(1), x(2) and x(3) = parameter values of the regression equation to be obtained by optimization using GA technique.

4.2 Model used for Optimization

The model used for optimization of the parameters is obtained by carrying out the Robust Regression analysis of the datasets. The model is:

Sobserved = α + b*x(1) + c*x(2) + d*x(3); (2)

Where, a, b and c are the model parameters.

Keeping these parameter values in the regression model, further optimization of these parameter values is being done using GA optimization technique so as to be able to improve the model parameters and hence enhance the prediction accuracy. Wherein the constant values, 'b', 'c' and'd' are optimized so as to lead to a better solution method. An effort is made to make the computed values of the development effort very close to the measured value, leading to a very root mean square error (RMSE).

Implementation and Results

For the present problem, the fitness function used is

Minimize Abs. (∑(Smeasured – Scomputed));

Where, S = software effort estimation value measured in man-months for tuning of model parameters. Where Smeas, is measured value of effort, Scomp is computed value of effort according to the model used. In order to minimize the total squared error given above, genetic algorithm is used changing the parameter values of the model. The code for the Objective function used is written as M-file in M-File editor and is recalled in the MATLAB command window. The lower and upper bounds of the three variables 'b', 'c' and 'd' as specified in the estimation model are fixed based on the values used in the linear regression model as given in equation (2) above.

5.1 Algorithm for Software Cost Estimation Framework:

The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:

• Selection rules select the individuals, called parents that contribute to the population at the next generation.

• Crossover rules combine two parents to form children for the next generation.

• Mutation rules apply random changes to individual parents to form children.

The following outline summarizes how the genetic algorithm works:

• The algorithm begins by creating a random initial population.

• The algorithm then creates a sequence of new populations. At each step, the algorithm uses the individuals in the current generation to create the next population. To create the new population, the algorithm performs the following steps:

1. Scores each member of the current population by computing its fitness value.
2. Scales the raw fitness scores to convert them into a more usable range of values.
3. Selects members, called parents, based on their fitness.
4. Some of the individuals in the current population that have lower fitness are chosen as elite. These elite individuals are passed to the next population.
5. Produces children from the parents. Children are produced either by making random changes to a single parent-mutation or by combining the vector entries of a pair of parents-crossover.
6. Replaces the current population with the children to form the next generation.
7. The algorithm stops when one of the stopping criteria is met.

Results and Discussions

After optimization of the fitness function using MATLAB command, the optimized function value and the optimal parameter values are obtained. Using different parameter options for GA algorithm functions solutions obtained are as follows:

$$S_{estimated} = S_{observed} = \alpha + b*x(1) + c*x(2) + d*x(3); \quad (3)$$

Where

α = 143.654790554716, a constant term

b = *0.802930099303848*

c = *0.212133114686023,*

d = *0.0811574301282941*

The lower and upper bound values used for the parameters are as follows:

lb = [0.75 0.15 0.07];
ub = [0.85 0.25 0.10] ;

Further the figure 3 below shows optimized parameter values of all the datasets using GA optimization.

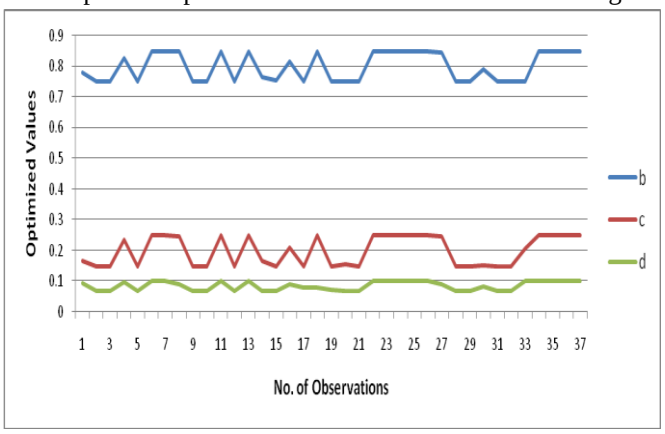


Fig. 3: Plot of optimized parameters "b", "c" and "d" for datasets

On further analysis of the above equations (2) and (3), it was seen that equation (3) for the model was found to be the best developed model, resulting in low RMSE value of 61.66 as compared to that of regression model for the same datasets having RMSE value of 96.31. Also, MAE values for RR and GA based models are 0.17188 and 0.098818 respectively, which again demonstrates the superiority of GA over other techniques (Table 2 and Fig. 4 & 5 below). Further, it clearly demonstrates that genetic algorithm optimization techniques have been successful in developing a better prediction model by lowering the RMSE value. The MATLAB plot of the various functions used in the optimization of the model has been shown in thesis. Further, the various fitness function values of parameters which is to be optimized has been plotted.

Table 2: RMSE & MAE values using RR & GA

	Using Regression	Genetic Algorithm
RMSE	96.31	61.66
MAE	0.17188	0.098818

Further, from the perusal of comparative plots of observed and predicted effort values as given in Fig. 4 & 5, both for RR and GA based, it is seen that for both RR and GA based model the predicted values closely follows the observed trend, but still GA based trend is almost superimposed over one another.

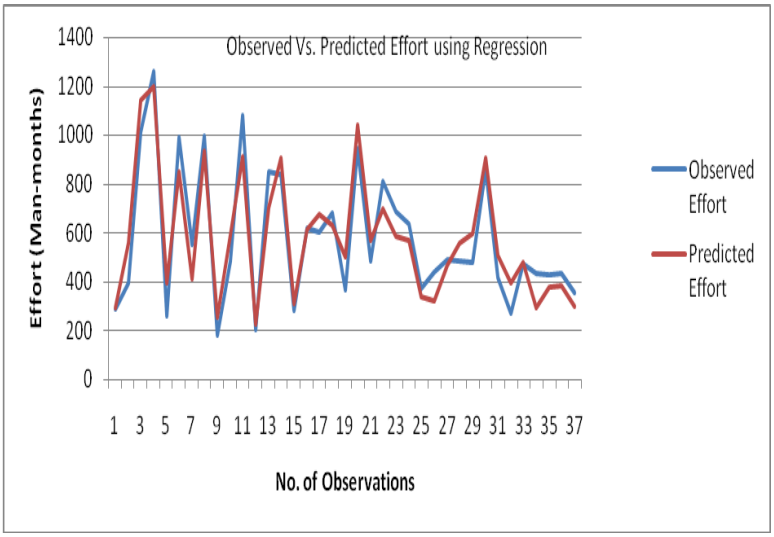


Fig. 4: Plot of Observed Vs. Predicted Effort using RR

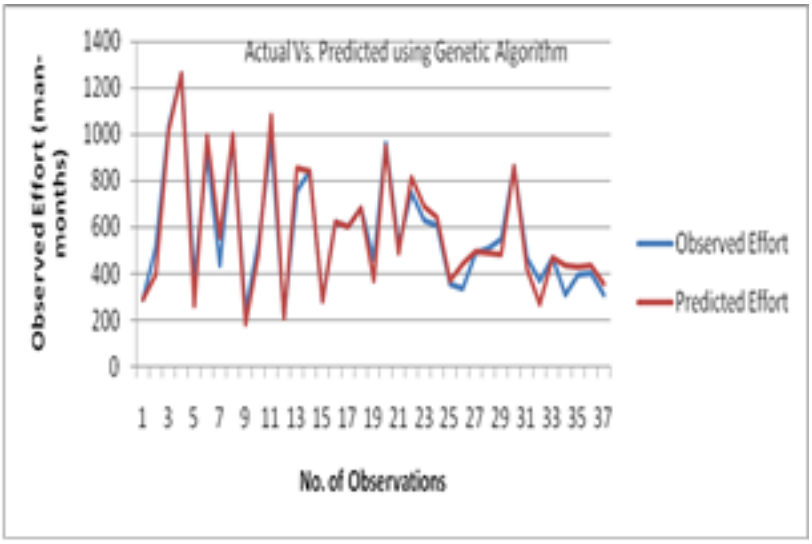


Fig. 5: Plot of Observed Vs. Predicted Effort using RR

Conclusion

In this study, applicability and capability of Genetic Algorithm techniques for application in software cost estimation as a predictive tool has been investigated. It is seen that GA models are very robust, characterised by fast computation, capable of handling the noisy and approximate data that are typical of data used here for the present study. From the analysis of the results given earlier it is seen that GA has been able to perform well for the prediction of effort estimation. Due to the presence of non-linearity in the data, it is an efficient quantitative tool. The studies have been carried out using MATLAB simulation environment.

In the present work a new model based on the robust regression model equation using genetic algorithm optimization technique has been developed, wherein the constant values, 'a', 'b' and 'c' are optimized so as to lead to a better solution method. An effort is made to make the computed values of effort estimation very close to the measured value, leading to an improved root mean square error (RMSE) and Mean Absolute Error (MAE). The data set that has been used consists of three independent object oriented variables obtained during the design phase, viz. Number of Services Requested (NSR), the Number of External Methods (NEM) and the Number of Attributes (NOA).

A GA model, using different options viz. Population, fitness, selection, mutation , crossover, hybrid and stopping criteria and their combinations has been developed for the prediction of effort estimation. From the analysis of the results given under the heading "Results and Discussions", it is seen that using the above options, an improved prediction model over the regression one has been developed, resulting in a lower RMSE and MAE values of 61.66 and 0.098818 respectively as compared to the earlier one from the regression model as 96.31 and 0.1718818 respectively.

References

[1] Mogili Umamaheswara Rao, et. al. (2014), "Effort Estimation for Object-Oriented System Using Artificial Intelligence Techniques", Volume No: 1(2014), Issue No: 10, pp. 248-252.

[2] Pushpendra K Rajput, Geeta Sikka, and Aarti, (2014), "CGANN-Clustered Genetic Algorithm with Neural Network for Software Cost Estimation", International Conference on Advances in Engineering and Technology (ICAET'2014), pp. 268-272.

[3] Farooq Azam, et. al. (2014), "Framework Of Software Cost Estimation By Using Object Orientated Design Approach", INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 3, ISSUE 8, pp. 97-100.

[4] N. Veeranjaneyulu, S.Suresh, Sk.Salamuddin3 and Hye-jin Kim, (2014), "Software Cost Estimation on e-Learning Technique using A Classical Fuzzy Approach", International Journal of Software Engineering and Its Applications Vol. 8, No. 11 (2014), pp. 217-222.

[5] Lalit V. Patil, et. Al., (2014), "Develop Efficient Technique of Cost Estimation Model for Software Applications", International Journal of Computer Applications (0975 – 8887) Volume 87 – No.16, February 2014.

[6] Jenna Carr, (2014), "An Introduction to Genetic Algorithm".

[7] Jyoti G. Borade, (2013), "Software Project Effort and Cost Estimation Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, pp 730-739.

[8] Divya Kashyap and A. K. Misra, (2013), "AN APPROACH FOR SOFTWARE EFFORT ESTIMATION USING FUZZY NUMBERS AND GENETIC ALGORITHM TO DEAL WITH UNCERTAINTY", Computer Science & Information Technology, pp. 57–66.

[9] K. Subba Rao, et. al. (2013), "Software Cost Estimation in Multilayer Feed forward Network using Random Holdback Method", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 10, pp 1309-1328.

[10] Simon, D. (2013). Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence. Hoboken: Wiley.

[11] Matlab 2012a: Optimization Toolbox - Product Documentation.

[12] Vahid Khatibi, Dayang N. A. Jawawi, (2011), "Software Cost Estimation Methods: A Review", Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No. 1, pp 21-29.

[13] Erik D. Goodman, (2009), "Introduction to Genetic Algorithms", 2009 World Summit on Genetic and Evolutionary Computation Shanghai, China.

[14] Martin Shepperd and Chris Schofield, (1997), "Estimating Software Project Effort Using Analogies", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 23, NO. 12,, PP. 736-743.

[15] Stein Grimstad, et. al, (2006), "A Framework for the Analysis of Software Cost Estimation Accuracy", ISESE'06, ACM 1-59593-218-6/06/0009.

[16] Andrey Popov, (2005), "User Manual, Genetic Algorithm for Optimization", Programs for MATLAB, ver. 1.0.