Turkish Journal of Computer and Mathematics Education Vol.12 No. 4 (2021), 615-624

Research Article

Software Cost Estimation Technique Based On Multiple Artificial Neural Network Models

Wathq Asmael Hamed

Directorate of Education in Nineveh, Mosul, Iraq

Wathiq8000@gmail.com

Article History: Received:11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract: Software cost estimation is an essential and important endeavor for the effective implementation of applications development project concerning its price & time plus its direction concerning its monitoring of autonomous applications development jobs. Software cost estimation is the prediction of software development endeavor and applications development time necessary to create a software job. The scheduling is of scheduling Resources, Budget, Time and several equally Precise software cost estimation is regarded as a tricky job as the information concerning the application project to be designed in the time of its beginning and completion remains obscure, thus drives the investigators from both professors and business to research in the exact same. What's more, it's always preferable for any approximation version to be inclusive because precision in estimation versions mutually lies together using their inclusiveness. So software cost estimation procedure being predictive in character hence requires for inclusiveness that will consequently bring inside that the precision. Within this paper, we'll present many versions for software cost estimation according to variants from Artificial Neural Networks which were completed within the research study. One of those models relies on exact choice of drivers as input into an Artificial Neural Network. And others derive from hybrids of Artificial Neural Networks with distinct Meta-heuristic algorithms as utilization of meta-heuristics in forecast issues such as that of program cost estimation is becoming more popularity. Everyone these versions have been experimented with variety of valid data collections.

Keywords: Software cost estimation, Artificial Neural Network, Meta-heuristic algorithms.

I. INTRODUCTION

Software cost estimation is that the calling of improvement effort and improvement period required needed to create a software job. It's regarded as the exact first measure of program development procedure and in precisely exactly the exact identical time regarded as the vital task as precise assessments of expansion of this present job, its own shipping exactness and its price control may only be attained once computed estimation is true. And at wider perspective a true estimation of a presently growing software project is going to lead to landing the business at a much better program of its autonomous applications jobs also. With due aforementioned mentioned rationale, application effort estimation has obtained a significant quantity of focus of several researchers from previous a long time. To put it differently, Software cost estimation is that the summation of forecasts of both construction campaign and calendar time useful to develop an application project. The building endeavor comprises the summation of operating hours and also the entire number of employees found in the task for gentle project development. In the start of program job development, institutions with the character arrived to the matter of inferior estimations of growth endeavor and development time of software tasks. An excellent cause for that really is and that's persistent this period is that the accessibility to obscure advice about the applying endeavor to be stated at some time of a unique estimation procedure. A greater estimate of software product is your only matter that might allow any application improvement job manager to speed the job progress, provides him or her great tabs on potential cost precision and control at shipping span. This in predominant, nevertheless, supplies the company a superior understanding of resource usage and can land the company at a significantly better program of a unique endeavors. With this particular usage, an Excellent variety of software cost estimation approaches from previous a lot of years have been suggested that range from algorithmic to noninvasive algorithmic to Information mining and also to meta-heuristics algorithmic established, but sadly non among these fulfill the approval standard of precision in estimation of software development jobs. But varying arrays of neural network based versions together with their hybrids also has been developed and have demonstrated improvements previously. Within this study, precision in software cost estimation is accomplished by developing quote model based on hybrid vehicle of artificial neural network plus a few meta-heuristic optimization calculations.

Even better software cost estimation models could be persuaded by a data set comprising significantly less, but highly pertinent features. Thus, within this subsection, software cost estimation is accomplished by first executing a proposed input choice process to acquire the appropriate pair of cost drivers leaving the insignificant features. Within another step, it's currently only these appropriate set of features which are being delegated to Artificial Neural Network as its input signal for the role of obtaining the precise estimation of software development effort and price. Eliminating the insignificant cost drivers in the very initial step leads to reach precise software cost estimation success. In the proposed model the issue of ascertaining the precise estimates of applications price is done with two measures. At the very initial measure, proper choice of input is completed which then functions as an input into an artificial neural network model characterized in second measure.

II. RELATED REVIEW

For the intent of creating a precise and trustworthy software cost estimation model, it wants a controlled method of using very fantastic number of time and resources. Such as the application development job, software cost estimation should also be handled as a job even though of a tiny scale character and thus as with other jobs has to be handled and handled carefully. Additionally, there are lots of associations with distinct software cost estimation procedures they follow along. These different procedures differ in many facets and not one of the 2 associations is found to adhere to the identical estimation procedure. In virtually all applications development businesses across the planet, a frequent thing discovered is that the inadequate estimation of growth effort and advancement which leads to vast type of glitches in most unsuccessful applications development jobs. Additionally different “business jokes" happen to be divided on estimation processes such as a typical one where applications development endeavor of any endeavor is assumed to equivalent to multiplication of operating hours to Pie and adding 30 more percentage for it. Actually however, it's common that some perform attempts are called less compared to their actual effort as suggested by [Haikala and Märijärvi, 1997]. Software development however a procedure of continuous improvement, begins with a really over all notion of this applications to be made. The remaining details that will land into some victory in program project estimation eventually become clear and nicer when the prerequisites of the software product has to be developed and its own corresponding aims become apparent also. This way the precision in estimation of applications project increases with the development of its job in its whole life cycle. In regards to inaccuracy in estimation procedure to be the motive of price growth and because of lack of concrete understanding regarding the applications project to be grown, Barry Boehm reports that applications price estimates therefore ready in the starting phases of software development procedure could be taken out by a variable four [Boehm, 1981].

The period computer program quality is generally observed at a lean logic and within an isolated way from the process of growth. Software quality consequently becomes limited to notions such as "usability", "mistake freeness", "extensibility", etc. Item quality thus dismisses both standard parameters of time and cost. Critical applications quality, that can in actual, be called as applications process quality, on the opposing side, must pay for those aspects also. Therefore, the very best applications concerning software quality procedure are often just great enough applications concerning quality of merchandise. This section is regarded as a service to key notion of applications quality and to quality of merchandise.

Software size and cost estimation is a forecast issue of very large magnitude. Software cost estimation is limited not just to software development firms but is enlarged to application’s to be produced at other areas like people of Research and Development associations, Aircraft associations and several equally. Because dimensions of applications project is generally regarded as the most dominant element in specifying software's price, very excellent size quotes of almost any software job size are useless to great price estimation. Slightly looking for the faultless way of estimating precise dimensions and price, a more real way to boost estimation is to decrease the dangers Involved with insufficient sizing and incorrect breaking of applications in parallel. The intention of this segment is to help proficient cost specialists in understanding the resources of vagueness and danger of danger in breaking and sizing, and also to provide insight to extenuating the dangers while building decisions regarding various costing and sizing choices. The belief of hazard will be dominant to some this sort of investigation, as well as the 2 techniques which have the capability to boost accountability of holding dangers associated with applications cost quotes includes identification of these zones of doubt that afterwards convert and move to dangers and extensive Analysis of entire procedure for software cost estimation procedure so as to regulate the regions where there is potential of decrease in vagueness concerning its risk reduction.

III. PROPOSED MODEL

Input Selection: Even better software cost estimation models could be convinced out of a data set comprising less, but exceptionally pertinent features. According to this, an overall selection process of inputs will be suggested. The selection process selects the appropriate project characteristics, also known as cost drivers using the aforementioned process on every individual data collection. The input procedure selects the applicable and dismisses the insignificant job characteristics using the frequent enumeration method of choosing one feature from one specific dataset, assigning this information as an input signal to an present artificial neural network based version and calculating the operation of an present neural network based method with this feature. This approach is repeated until all probable combinations of characteristics are analyzed over precisely exactly the exact identical version to eventually receive a pair of features which lead to a desirable estimation of program development expenditure to be determined by means of a threshold set concerning magnitude of relative error (MRE) value gained from such types of combinations. Inside this study three datasets are employed and 2 methods are preceded employing exactly the exact identical procedure. The Input selection process is given on following page:

Algorithm: Pseudo code of Proposed Input Selection Procedure ( )

1. For any data Set S

2. For (i = P; i> 1; i--; P= Number of attributes in D

3. do

4. For (k = 1;k<i;k++)

5. do

6. Delete k^{th} attribute.

7. For (l = 1; l<r; l++); r=Number of techniques used

8. do

9. Assign the input to a model without the discarded attribute “k”.

10. Estimate the performance of the model for the given dataset.

11. End

12. Calculate the mean performance of all the above, “r” technique under this discarded attribute of P.

13. End

14. Create the matrix comprising "N" rows and 3 columns representing title of every method as "P" (feature) occasions, their corresponding ordinary performance and numerical corresponding lost attribute.

15. End

16. Create the matrix Comprising P*n rows and their corresponding ordinary performance and their numerical count of lost 3columns symbolizing the title of every method as "P" occasions, attributes.

From the aforementioned proposed algorithm, initial a data collection "P" is obtained with all its features as shown in steps 2 and 1. In steps 7 - 11, every technique is worked to a dataset (dataset without feature "k") to automatically figure out the functioning of the procedure with no lost attribute. These measures are repeated dependent on the amount of processes. In measure 12 calculation of typical performance of the "r" methods beneath this elimination of jth feature is performed. Measures 3-15 are replicated for several values of "k" and each of the time distinct versions consequently will be worked using distinct values of "j". Each of the aforementioned works for"P" variety of features which then is decremented for P and P-1th feature and then to N, P-1 along with P-2 F feature and so

forth. This can be controlled with the outside "Id" loop. At the conclusion of the algorithm info about demonstrations of different versions under distinct subsets of features implemented on a frequent data collection is accomplished. More importantly, by the achieved information, accurate differentiation between the appropriate set of characteristics and insignificant set of features is completed and thereby elimination of irrelevant characteristics is achieved while going for your estimating procedure.

Artificial Neural Network Model: The operation of almost any artificial neural system is characterized by its fundamental architecture including various parameters such as number of hidden layers in neural system and also the neuron (node) rely on each one of those layers, move function used at every node, weights and parameters of this training algorithm employed for example their configurations also. Additionally, network performances have been significantly determined by the proper choice of network routines and related learning principles. A large challenge is to produce decision when picking quantity of volunteers in each layer, number of layers and also the choice of training algorithm. But, it's always preferable to choose minimal possible number of nodes and layers at every layer with no compromise on functionality. At step1, 17 price drivers have been decreased to 11; these 11 drivers ' are subsequently regarded as input into the planned artificial neural system. These inputs are provided weighted inputs by joining a weight with each enter signal and calculate the attempt together with the following equation 1:

$$\log(E) = \log(\mathrm{a} * [\mathrm{size}]\mathrm{b}\ \mathrm{X}) \prod_{\mathrm{i}=1}^{10} \mathrm{EMi} \qquad (1)$$

The output signal obtained with Formula 1, is contrasted with all the activation function and the output is forwarded. Dependent on the value taken from this activation function, the weights employed on the inputs have been all tailored. When the output in the stimulation function becomes a difference of calculated and actual campaign is considered and when found to be within tolerable, the outcome signal is approved otherwise the weights have been corrected. This way one complete iteration of this job is finished called epoch of this undertaking.

Artificial Neural Network & Hybrids of Meta-heuristic Algorithms: Software cost estimation is an optimization issue. A hybrid version of an artificial neural system & Because there is a rise in complexity of marketing issues and in precisely exactly the exact identical time collapse of algorithmic versions in solving these meta-heuristics are gaining attention due to their capacity in solving complex marketing issues and in precisely exactly the exact identical time attaining accurate answer. Within this subsection we'll discuss a variety of bases of meta-heuristic algorithms and artificial neural networks which were completed during this study and we'll see their effective usage in resolving the estimation problem of software development expenses. The Many versions used in our analysis include

A hybrid version of a useful connection artificial neural network & enhanced particle swarm optimization algorithm.

This meta-heuristic algorithm utilizes some basic parameters whose values can easily be adaptable. In addition, the convergence rate concerning locating the universal optimized option of those meta-heuristic calculations is much superior to algorithmic versions. These calculations hold evaluation and searching specification and consistently attempt the issue before an optimized solution can be attained.

Functional Link Artificial Neural Network Model & Improved Particle Swarm Optimization: Within this analysis, applications development effort and growth period is projected by deciding upon a functional connection artificial neural system. An operating connection artificial neural network is a premier string, single-layer feed forward artificial neural network. It consists of coating to carry input, called input signal and also a coat to get dispatching output, called output signal. Moreover, it features no hidden layers as the full processing must be performed on inputs in hidden layers is basically achieved with extension before being hauled in to the outer lining. Each input neuron keeps a feeling to part of input vector. The output signal on the alternative hand includes just one output neuron that calculates the application development endeavor like a linear weighted sum of the sparks arising out of the input. Due to the fact the MLPs usage straight back propagation algorithm as of the training algorithm, in addition being multi-layered that they face really a slower training speed struggles. Hence, that the non-availability of hidden layers through this system arrangement decreases the computational sophistication and omits the above issues. In addition, operational connection artificial neural network enriches the representation of input signal, thus allows it to furnish a larger precision of applications cost estimation and owing to its uniformity, it'll the usable expansion readily using an instant convergence speed. A lot of scientific tests are performed on Channel equalization, System analysis, Noise investigation and also a ton more through the use of functional connection artificial neural networks. These studies have shown a significant positive influence within their own outcomes.

Architecture of Proposed Functional Link ANN: Within our suggested operational connection artificial neural network (FLANN), so as to gauge software development endeavor, we ascertain the system design parameters in line with the features of constructive cost version II. An overall structure of operational connection artificial neural system is displayed at figure 1.

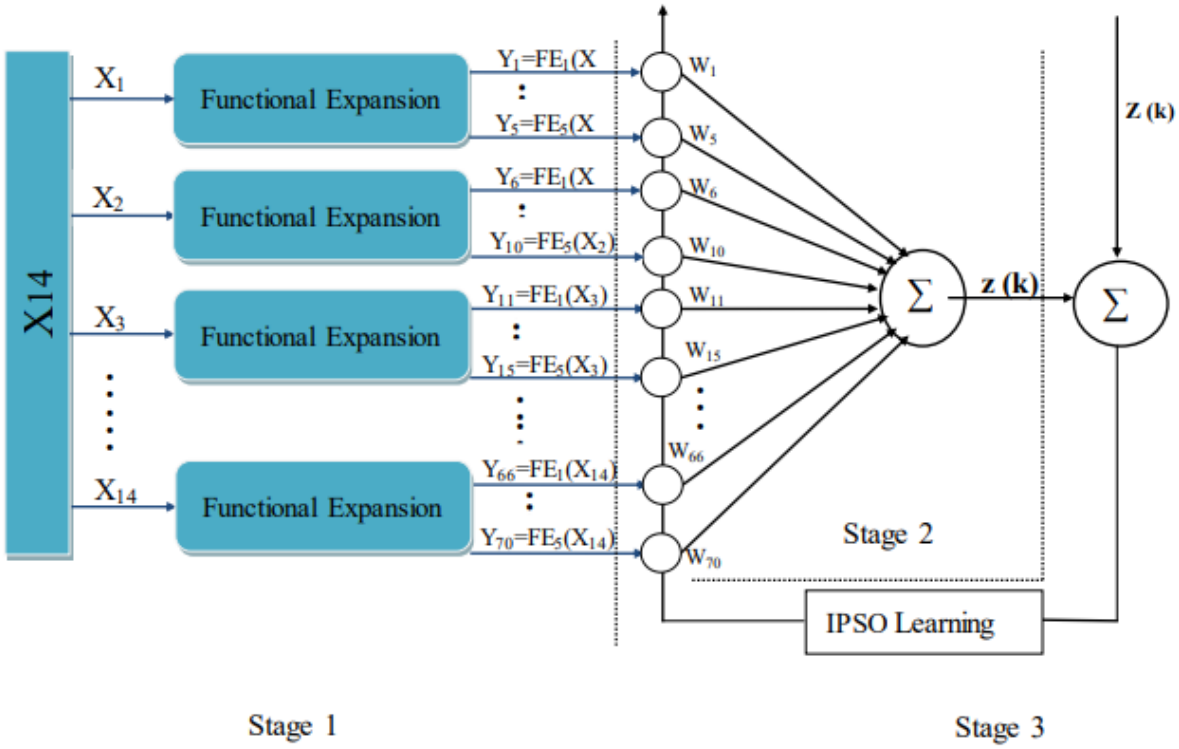

Figure 1: The Functional Link Artificial Neural Network Architecture

From the FLANN shown previously in figure two, "q" signifies the input-output pairs to be heard from FLANN. The input vector "Yq" consists of n size along with also the output "Zq" is scalar. Therefore, undefined signify the training routines. So as to functionally extend the enter Yq=[y1(q) y2(q) ••• yn(q)]T, a group of N basis functions Ø(Yq)=[Ø(Yq) Ø(Yq) ••• Ø(Yq)]T are utilized. It's these N linearly independent functions that are responsible for mapping of n-dimensional distance into N-dimensional space. This mapping is given as:

$$R_n \rightarrow RP,\ p<P$$

These functions may be linearly mixed in a relational type as X WØ. Since Xq = [X1(q) X2(q) ••• Xm(q)]T, in which W is the m×N weight matrix. The complete output Z= [z1 z2 ⋯ zm]T, zj= ρ(Xj), j =1, 2, •••, m is made by carrying the matrix Xq via a pair of non-linear function ρ(•) = tanh(•). The different steps involved here will be:

Step 1: An input signal of 14 cost variables (Input motorists) of this supported dataset is supplied to the system that's subsequently enlarged natively with Chebyshev, Legendre and Power Series functions.

Step 2 Let b (I), 1 < I < X signifies every component of the input prior to growth. Next each b (I) is expanded to bn (I), 1 < n < N (where N = number of enlarged factors for every input component. In our analysis, N is X and 5 represents the entire amount of attributes of their selected dataset. Expansion of every input could be awarded a

$$y0\ (z(i)) = 1,\ y1(b(i)) = b(i),\ y2(b(i)) = 2b(i)2 - 1,$$

$$y3(b(i)) = 4y(i)3 - 3y(i),\ y4(b(i)) = 8y(i)3 - 8y(i)2 + 1.$$

Where, b (I), 1 < I < d, d reflects the attributes set from the dataset. Each of the nonlinear presses are then multiplied by numerous initialized weights selected randomly in the range [-0.5, 0.5] then summed-up to create the projected output signal (k). Each of the z (k)'s are eventually summed to find the Z (k).

Proposed Improved Particle Swarm Optimization (IPSO): The two large flaws of particle swarm optimization specifically inefficiency in good direction of slow and easy looking for finding the world wide most effective compels us to select Improved Particle Swarm Optimization which is a standard global form of an Particle swarm optimization. Improved Particle Swarm Optimization more lucrative as after every production, the funniest particle with the present generation was replaced with the exact ideal particle of their last creation ergo leads to the achievement of accurate results. In current research [Eberhart, R.C., Shi, Y., 2000] a variety of selection methods for picking inertia weight have been indicated. In this implied Improved Particle Swarm Optimization process early weight is diminished quickly from the very initial stages while at best afterwards, the very first weight is diminished slowly. The Critical process to Coach indicated operational relationship artificial neural network will be supplied as follows.

PROCEDURE

$Input: Initial\ weight = \mathrm{v0};$

$Linear\ section\ (End\ point) = \mathrm{v1};$

$Number\ of\ generations = G1$

$Max\ number\ of\ generations = G2$

$reduce\ \mathrm{v}(\ )$

$begin$

$for\ q = 1; Gen1\ \mathrm{v1} = \mathrm{v0} - (\left(\frac{\mathrm{v1}}{\mathrm{G1}}\right) * \mathrm{k})$

end

$for\ k = G1 + 1 : G2$

$\mathrm{v1} = (\mathrm{v0} - \mathrm{v1}) * \exp\left(\frac{(\mathrm{G1} + 1) - \mathrm{k}}{\mathrm{k}}\right)$

end

end

the values of G1 and G2 are based on

emperical knowledge

IV. RESULTS AND DISCUSSIONS

This subsection introduces the experimentation effects of this suggested technique mentioned. In this subsection we will very first demonstrate the results obtained while implementing the program cost estimation model based on hybrid input selection process and artificial neural network version. We will demonstrate the got experimentation consequences of software cost estimation model according to operational connection artificial neural network and also enhanced particle swarm optimization. And finally we'll finish this subsection using the outcomes of program cost estimation model based on artificial neural network and firefly algorithm. The MRE worth of software cost estimation model based on hybrid input selection process and artificial neural system model is calculated for randomly chosen set of jobs from 2 datasets from four cited in the information collections subsection and compared with the results obtained utilizing COCOMOII version that's regarded as the simple estimation model in applications

development jobs and also another present version by [Tirimula Rao, B., 2009]. Fewer than two tables specifically Table 1 and Table 2 demonstrate that the significant gap in MRE of this suggested technique against both existing methods.

Table 1: Magnitude of Relative Error (%) of Proposed Model of COCOMO81 Dataset

S. No	Magnitude of Relative Error (percent) ON CONSTRUCTIVE COST MODEL DATASET		
	COCOMO II Model	B.T.Rao et al Model	Proposed model
1	7.42	5.21	3.41
2	19.81	9.08	7.06
3	6.42	3.18	2.20
4	50.92	14.40	11.09
5	12.36	4.68	3.69
6	5.34	3.49	2.67
7	16.42	7.48	4.01
8	8.62	3.18	2.87
9	13.08	8.56	4.08
10	6.12	3.21	2.76

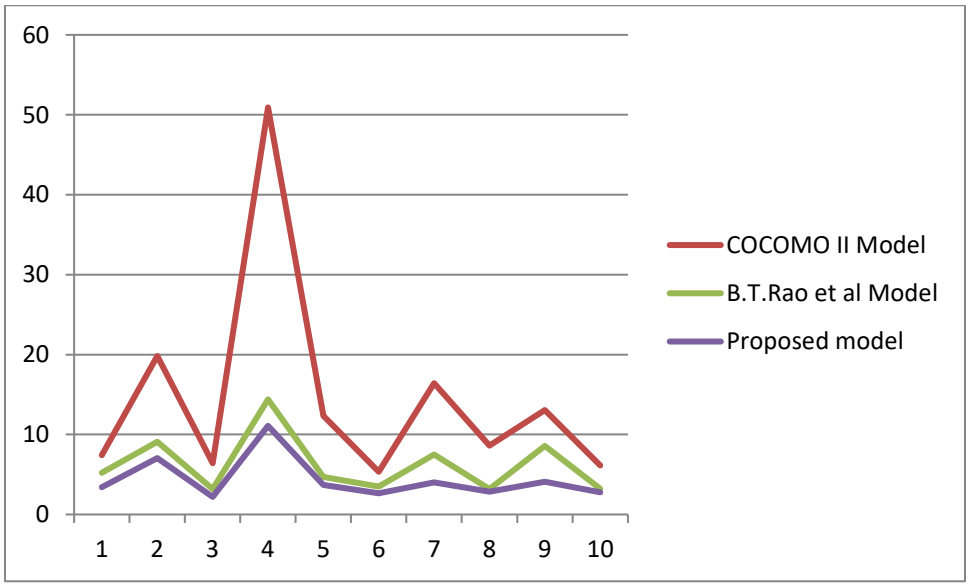

Figure 2: Magnitude of Relative Error (%) Based Graphical Description and Proposed Model on constructive cost model Data Set

Table 2: Magnitude of Relative Error (%) of Proposed Model and COCOMO81Dataset.

S. No	Magnitude of Relative Error (%) on COCOMO81 Dataset	
	COCOMO II Model	Proposed Model
1	7.42	3.48
2	19.81	6.08
3	6.42	3.08
4	50.92	14.08
5	12.36	4.12
6	5.34	2.58
7	16.42	3.0

8	8.62	2.56
9	13.08	3.88
10	6.12	2.67

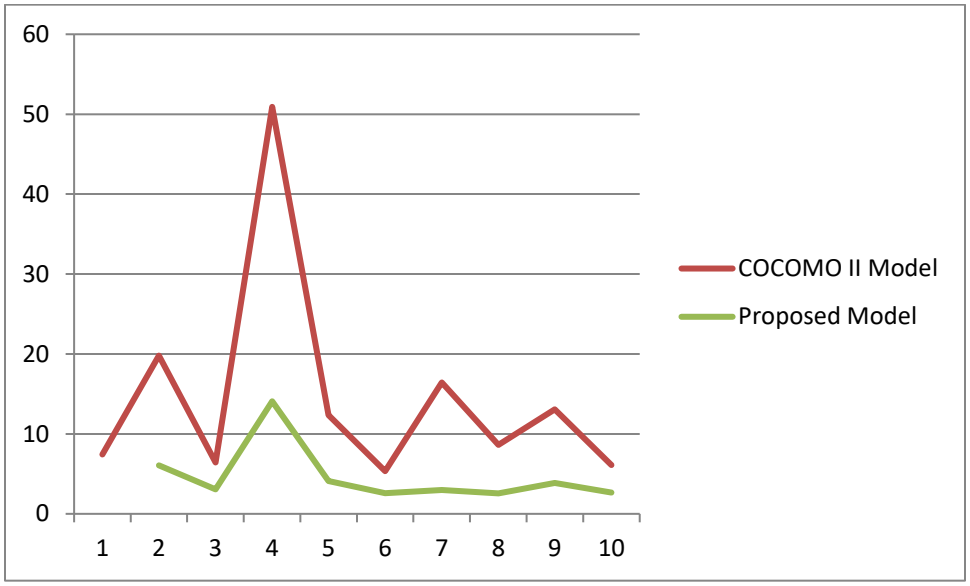

Figure 3: Graphical Description of Magnitude of Relative Error (%) of COCOMO and Proposed Model on COCOMO Data Set

The MRE worthiness of software cost estimation model based on hybrid input choice process and artificial neural network design is calculated to get randomly chosen group of jobs from 2 datasets from four cited in the information collections subsection and compared with the results obtained utilizing COCOMOII version that's thought of as the simple estimation model in applications development jobs and also another present version by [Tirimula Rao, B., 2009]. After implementing the suggested software cost estimation model according to operational connection artificial neural network (FLANN) and enhanced particle swarm optimization (IPSO) algorithm, then here within this subsection we'll show the obtained outcomes both in tabular in addition to in graphic form. The MRE worth of suggested version is calculated to get randomly chosen set of jobs from all the four datasets as specified from the datasets subsection. The outcomes are then contrasted with the results gained utilizing COCOMOII version that's regarded as the simple estimation model in applications development jobs and with yet the other present version by [Tirimula Rao, B., 2009]. Hence according to their own MRE values, graphic demonstration as displayed in Figure 4 and also can be utilized to illustrate the operation of two proposed and existing version when implemented on COCOMO 81 dataset.

V. CONCLUSION

The principal concern of any application merchandise customer would be to secure a facility which can satisfy its operational needs, of the mandatory caliber, and delivered in a decent budget and period. Price estimates prepared in the first phases of a program development jobs enable the customers to carry out a cost-benefit evaluation, secure financing in addition to used as a foundation for price control during delivery. Where the program product is a commercial advantage, the first capital investment has to be balanced with the expense of operations and maintenance within the life-time of their application product to make certain the project stays rewarding and planned yields of capital expenditure have been attained over an estimated interval. Decisions made in the first phases of the program development project consequently carry far-more attaining economic implications and will seal the fiscal destiny of almost virtually any software development company. True software cost estimate is that the condition of art of their applications engineering tasks and naturally it's an intricate procedure. It's well known that the precision of the person software effort estimation models could be described based upon comprehending the calibration of their applications data and it was verified using hybrid estimation versions completed in this research. This paper concludes this estimating software development expenditure by simply introducing the two-step procedure of

suggested input and artificial neural system process yields greater functionality as the addition of input selection process causes a model that's more secure since budding co-linearity between characteristics is diminished.

VI. FUTURE SCOPE

Researchers to the progression of applications price forecast systems have recognized the simple fact there are lots of choices to other and algorithmic nonparametric of software cost estimation which might comprise data mining into machine learning versions. But for the powerful potentiality of at least one of these methods to be accomplished it's essential that they're brought closer to this quote practitioner. By way of instance, in the event of neural networks there's a specific amount of proficiency required before they may be deployed efficiently by a professional as it will become complicated with regard to their own trainings as well as other architectural intricacies to embrace them that amounts to their lacking of desirable competence. The identical strategy might be protracted to bargain with the technology-specific problems that are essential for the current day world with technologies that are abundant for any kind of alternative. The suggested approach may also be extended for creating project-specific effort estimate models that might be more suitable for number of jobs, varying concerning character of program, dimensions and relevant facets. So, there's sufficient scope for further expansion of this suggested idea.

REFERENCES

1. Albrecht, A. J., “Measuring application development productivity”, Proceeding of the Joint SHARE, GUIDE and IBM application development symposium, IBM Corporation, 1979.
2. Albrecht , A.J., and Gaffney, J.E., “Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation,” IEEE Trans. Software Eng., vol. 9, no. 6, pp. 639-648, Nov. 1983.
3. Bardsiri, A. K., & Hashemi, S. M.,“Software effort estimation: A survey of well-known approaches”. International Journal of Computer Science Engineering (IJCSE), 3(1), 46-50. 2014
4. Brooks, F.,“The Mythical Man-Month”,Addison-Wesley Reading, MA, United States, 1975.
5. Chen,Z., Menzies, T., Port, D., and Boehm, B., “Feature Subset Selection Can Improve Software Cost Estimation Accuracy,” ACM SIGSOFT Software Eng. Notes, vol. 30, no. 4, pp. 1-6, 2005.
6. Chiu, N.H., Huang, S.J., “The adjusted analogy-based software effort estimation based on similarity distances,”Journal of Systems and Software 80(4), 628–640, 2007
7. Dehuri, S., Cho, S.B., “A comprehensive survey on functional link neural networks & an adaptive PSO–BP learning for CFLNN”Neural Computing & Applications pp. 187-205, 2010
8. Eberhart, R.C., Shi, Y., “Comparing inertia weights and constriction factors in particle swarm optimization,” In: Proceedings of the 2000 Congress on Evolutionary Computation. Vol. 1, pp. 84–88, 2000
9. Fenton, N.E., Pleeger, S.L., “Software Metrics: A Rigorous andPractical Approach”, second ed. PWS, Boston, MA, USA, 1997
10. Gray, A. R..,MacDonnell, S.G., “A Comparison of Techniques for Developing Predictive Models for Software Metrics”, Information and Software Technology,vol. 39, 1997
11. Haikala, I., and Märijärvi, J., “Ohjelmistotuotanto, Suomen Atk-kustannus O”, 1997
12. Haugan, G., “Effective work breakdown structure”. Project Management Institute, 2001.
13. Idri, A., Abran, A. and Khoshgoftaar, T. M., “Fuzzy Case-Based Reasoning Models for Software Cost Estimation”, Soft Computing in Software Engineering, SpringerVerlag, 2002.
14. Jensen, R., “An Improved Macrolevel Software Development Resource Estimation Model”, Proceedings 5th ISPA Conference, pp. 88-92, April 1983.
15. Jiang, M., Luo, Y.P., Yang, S.Y., “Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm,” Information Processing Letters 102 (1), 8–16., 2007
16. Kitchenham, B., Pickard, L.M., Linkman, S., Jones, P.W., “Modeling software bidding risks”,IEEE Transactions on SoftwareEngineering 29 (6), 542–554, 2003
17. Li,J., and Ruhe,G.,“A Comparative Study of Attribute Weighting Heuristics for Effort Estimation by Analogy”. Proc. ACM-IEEE Int‟l Symp Empirical Software Eng., Sept. 2006
18. Miyazaki, Y., Terakado, Y., Ozaki, K., Nozaki, N., “Robust regression for developing Software estimation models”, Journal ofSystem and Software 27 (1), 16–35,1994
19. Menzies,T., Chen, Z., Hihn,J., and Lum,K., “Selecting Best Practices for Effort Estimation,” IEEE Trans. Software Eng., vol. 32, no. 11, pp. 883-895, Nov. 2006.

20. Yogesh Hole et al 2019 J. Phys.: Conf. Ser. 1362 012121
21. NASA, “Handbook for Software Cost Estimation”NASA, Jet Propulsion Laboratory, Pasadena, California, 2003.