Fuzzy Optimization Algorithm for Software Cost Estimation

Siva Suryanarayana Ch., Satya Prakash Singh

Abstract: *Estimation of a software cost depends on a probabilistic model and thus it doesn't create precise values. In any case, accessibility of good chronicled information combined with a efficient technique can create improved outcomes. This paper, we have displayed a Software Effort Estimation Model utilizing PSO and Fuzzy Logic. Fuzzy sets have been utilized for displaying uncertainty and imprecision in estimation of effort while PSO has been utilized for tuning parameters. This has been seen from the outcomes that Fuzzy-PSO intelligence gives precise outcomes when compared through its different partners. This system relies upon thinking by linguistic quantifiers and fuzzy logic. This kind of model holds well, when the product plans are communicated by absolute or potentially arithmetical data. Along these lines, this projected methodology improves the old style correlation process that doesn't think about clear cut data. In the fuzzy correlation model, fuzzy sets are used to describe both the clear cut and the arithmetical data.*

Keywords: Fuzzy Logic, Fuzzy Optimization, Particle Swarm Optimization (PSO), Software Effort Estimation, Software Cost Estimation (SCE).

I. INTRODUCTION

The Cost of Software Estimation is the way toward anticipating the measure of time essential to construct a software. The time is restrained as far as Person-Months (PM's) which is later on changed over into money cost. The essential contribution for the cost model is size dignified regarding KLOC (Kilo Lines Of Code) and systematic of Cost parameters. The benefit of cost estimation is Cost advantage investigation, legitimate asset usage (software, equipment and individuals), staffing plans, usefulness exchange offs, dangers and modify budget.

Other than the size of software, there are numerous other cost influencing factors. The most far reaching conventional of cost components are projected and utilized by Boehm et al in the COCOMO II model. These cost components can be divided into 4 sorts:

Product factors: Reliability; Dependability; reusability; item complexity; size of database utilized; documentation counterpart to life-cycle needs;

Computer factors: imperative time execution; principle stockpiling limitation; PC turnaround requirements; stage unpredictability;

Personnel factors: expert capacity; application experience; programming ability; stage understanding; language and device experience; work force congruity.

Project factors: development at multisite; software instrument utilization; development plan required. The overhead variables are not really autonomous, and the popular of them are difficult to evaluate.

In numerous models, a portion of the elements show up in joined structure and some are just overlooked. Additionally, a few variables take isolated values, bringing about an assessment work with a piece-wise structure.

The forecast of work effort and blueprint required for the progress and additionally upkeep of software structure assumes an indispensable job in software project management and this procedure is named as software cost estimation. The assessment of cost and time help in bristly approval and perception of the venture's headways at the hour of improvement process. The enlargement procedure is done; these assessments aid the calculation of venture efficiency. Software cost estimation can be characterized as the technique for anticipating the effort expected to shape up a software system. Dominant part of the cost estimation models try to develop an effort gauge, which would then be able to be altered into the undertaking time frame and use. Notwithstanding when cost and effort have adjacent affiliation, they are not essentially associated by a straightforward change work. Effort is commonly dignified regarding individual months of the software engineers, investigators and undertaking administrators. This effort gauge can be then be changed over into a money cost figure finished the calculation of a normal compensation for every unit time of the work connected and afterward, duplicating the resulting by the evaluated required effort.

Where the price is made to acceptable into the economical forced on the software venture. Therefore, it might be fundamental to reuse the estimations of different phases some times, decline the effort and obtainments or speculation more hazard to acceptable into the required budget. In the event that the timetable turns out to be excessively huge, costs will get raised in light of the fact that effort moves out to increasingly costly years The objective of software costing is to exactly conjecture the cost engaged with building up the software. In the event that the venture cost has been registered as a bit of an undertaking offer to client, at that point a decision must be made about the value referred to the client. Commonly, cost is just cost in addition to benefit. The estimation of Cost stays a composite problem, which keeps on pulling in significant research consideration. Researchers have endeavored various methodologies.

* Correspondence Author

Siva Suryanarayana Ch.*, Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, of Campus, Noida, India. cssuryanarayana@bitmesra.ac.in

Satya Prakash Singh, Department of Computer Science and Engineering Department, Birla Institute of Technology, Mesra, Ranchi, of Campus, Noida, India. sp.singh@bitmesra.ac.in

As of late, models dependent on AI model have been created. For instance, Finnie and Wittig applied case-based thinking (CBR) and artificial neural networks (ANN) to effort estimation. Utilizing an informational index from the ANN, Australian Software Metrics Association (ASMA) had the option to estimation development effort inside 25% of the candid effort in over 75% of the all projects, and with a MARE of below 25%. In any case, the outcomes from CBR were less promising. In 73% of the projects, the estimations were inside half of the actual effort, and for 53% of the projects, the estimations were inside 25% of the genuine.

II. LITERATURE SURVEY

There are many cost estimation models. The most extensively employed model was Constructive Cost Model. Kaushik et al (2012) have suggested the employment of back propagation method using neural networks for software cost estimation. The model hired identity function at the input layer and sigmoidal function at the hidden and output layer. The model integrates COCOMO NASA2 dataset and COCOMO dataset to trainer and to check the network. The test outcomes from the trained neural network were compared with that of the COCOMO model. From the experimental results, the combination of the conservative COCOMO model and the neural network methodology develops the cost estimation precision and the projected cost could be very close to the actual cost.

Satapathy et al (2013) in finding out the success or failure of the product the estimation of effort engaged in developing a software product performs an imperative role. Project managers need a dependable approach for software effort estimation. Satapathy et al (2013) have suggested a model which is executed by means of Multi-Layer Perceptron (MLP) and Radial Basis Function Network (RBFN) and produced results has been compared. In addition, a relative study of software effort estimation by means of MLP and RBFN has been offered. The results reveal that MLP model provides less value of MMRE, NRMSE and higher values of prediction precision. Therefore it was concluded that the effort estimation by means of MLP model will offer more precise results than RBFN.

Patil et al (2014) have offered a new idea where Principal Component Analysis (PCA) works with Artificial Neural Network (ANN) by possession the base of Constructive Cost Model II model. Feed forward ANN employs delta rule learning method to coach the network. Coaching of ANN was built on COCOMO II and PCA sample dataset repository. PCA was a kind of classification method, which could strain various input values into some definite values. It also assists in decreasing the gap between the estimated and actual effort. The suggested technology raises the perfection of the assessments without deteriorating the changeability, which develops the precision, reversal time and presentation of the system.

Software effort assessment was an imperative facet which comprises amount of effort, schedule, cost and manpower requirement. An Artificial Neural Network (ANN) forecast model that integrates with Constructive Cost Model (COCOMO) has been suggested by Dan (2013) which was enhanced by using Particle Swarm Optimization (PSO), PSOANN- COCOMO II, to offer a method which could assessment the software development effort precisely. In addition, it keeps the benefits of COCOMO model, developing the precision of software cost estimation and declining the fault rate. The results points out that, compared with original ANNCOCOMO II model, the precision of software cost and effort estimation has raised 3.27% by using PSO-ANN-COCOMO II model.

Rahul Kumar Yadav and S. Niranjan (2017) have proposed software effort estimation algorithm that has decided on fuzzy rationale for the cost estimation. The proposed technique has utilized three enrollment functions, for example, triangular, trapezoidal and ringer. A great deal of parameters including effort multiplier, Balance Relative Error (BRE), Absolute Relative Error (RE), Variance Accounted For (VAF), and so on have likewise been processed to look at the outcomes. Three arrangement of guideline sets have been utilized to assess the fuzzy result. The outcomes have demonstrated that every one of the three enrollment functions shift by 1-2% in the effort estimation. While, whenever contrasted and COCOMO-II, effort changes by 5-10% from the fuzzy yield. Ebb and flow research work has opened up a great deal of viewpoints for the future research laborers. The dataset that has been utilized in this system has been adjusted from NASA. Different datasets can likewise be attempted and tried.

Tarig Ahmed Khalid (2017) has proposed a fuzzy model to survey the cost of deformities evacuation at the beginning periods of venture arranging. This has been completed by considering the functional size of the venture and the experience of the development group. The proposed model has expanded the accuracy of the task cost estimation process. As indicated by Muhammad Ibrahim (2016), a few software projects have been created and delivered by software organizations yearly. Progressing ecological and mechanical advancements, cost restrictions, fumble, absence of aptitudes of the task bosses and off base estimation have made a large number of these plans bomb by and by. In huge software projects, SDCE has consistently been one of the main problems for the task manager and venture development group. The creators have proposed that the assets required are normally decided by individual month. There is an immediate identicalness between the measure of effort and the time required to finish a venture. In this way, the essential advance in SDCE is deciding the measure of effort expected to finish projects. In this paper, creators have utilized the bat algorithm (BA) to estimate the effort coefficients precisely and to lessen MMRE. Assessment has been directed with the NASA dataset having values from 60 unique projects. The outcomes have indicated that BA has significantly less error contrasted and COCOMO model.

Anupama Kaushik, et al. (2016) have incorporated a Firefly Algorithm (FFA) in ANN models for precisely estimating the SDCE. The FFA is one of the current developmental computing patterns, which is impacted by the organic conduct of fireflies. The ANN models that have been utilized in this method, has included functional connection artificial NN and radial predisposition function NN. Results have indicated that reconciliation of FFA with ANNs prompted a superior arrangement towards SDCE. Nazeeh Ghatasheh, et al. (2015) have proposed a firefly algorithm (FFA) for enhancing the software effort models.

The proposed FFA has been utilized for streamlining the parameters of three COCOMO-based models. The consequences of the FFA have indicated an improved performance over customary GA and PSO.

III. PROPOSED METHODOLOGY

A. Methodology

The uncertainty about cost estimation is generally exceptionally high, in perspective on forecast of basic segment size, cost drivers and different parameters. By displaying a couple of changes in the interim sort 2 fuzzy logic we can governor the uncertainty. In the current work fuzzy sets are used for exhibiting uncertainty and imprecision in a compelling way. The contributions of the standard cost model join an estimation of adventure size and evaluation of the parameters, instead of a single number, the size of software can be seen as a fuzzy set elastic the cost estimate moreover as a fuzzy set. We emphasize a strategy for engendering of uncertainty and ensuring infringement of the consequent effort. Fuzzy sets make an inexorably adaptable, high adaptable development condition. We make information furthermore the consequent uncertainty of the results.

The Parameter tuning is done by exhausting Particle Swarm Optimization. For each molecule position with values of tuning parameters, wellbeing limit is evaluated with an objective to limit the health work. The objective is to limit or expand wellbeing work. The particles moving towards ideal parameters by finishing a couple of accentuations until particles exhaust or backup of speed ends up being very nearly zero then we get the ideal parameters which are future used for effort estimation .

B. Proposed Model

The Proposed model involves three imperative sections. First portion is fuzzification process which perceives the sensible ending interims for the information parameter size. Second portion is parameter tuning using molecule swam upgrade. Finally, estimation of effort done through weighted ordinary defuzzification manner using the outcomes got in first and second steps.

i) Fuzzification process

The input size is fuzzified by using two sets of triangular fuzzy vales. The Triangular part limit is showed up

$$\mu_p(size) = \begin{cases} 0 & size \leq L \\ \dfrac{size + L}{2L} & -L \leq size \leq L \\ 1 & size > L \end{cases}$$

where L is the Mean of input sizes after fuzzification of the instructive file find the disguised regions (front) of the left hand side and right hand side those are called significant lower and furthest points of confinement of the enlightening file - Foot Print of Uncertainty. The methods for the Foot Print of Uncertainty are ending interims.

ii) "Parameter tuning using Particle Swarm Optimization"

The effort equation we considered is Alaa F .Sheta model-2
Effort =a* (Size) b +c*(ME) + d (10)

The parameters a, b, and c of the equation 10 are tuned by using molecule swarm optimization with idleness weight and MMRE as the wellness work (limit).

iii) "Defuzzification"

The defuzzification is done through weighted average method is as shown below
E= {w1 *[(a * α b)+c*(ME)+d]+ w2 *[(a*mb) + c*(ME)+d]+w3*[(a* βb)+ c*(ME)+d]}/w1+w2+w3
where wi is the weighting factor and α, m, and β are the fuzzified sizes obtained from triangular member function [15].

iv) Proposed work

The primary objective of our exploration is to utilize idea of soft computing especially fuzzy logic with COCOMO II for accomplishing precise software effort estimation and lessen the uncertainty in COCOMO II model. Engineering of proposed fuzzy model is as below The proposed fuzzy model has for input as Size of the task, and 17 cost drivers whose values are qualitatively characterized as very low, low, nominal, high, very high, and amazingly high conceived in to two groups relying upon their effect on the effort estimation as OG Optimistic group The range values of these multipliers are contrarily corresponding to effort and PG Pessimistic group The range values of these multipliers are legitimately relative to effort.
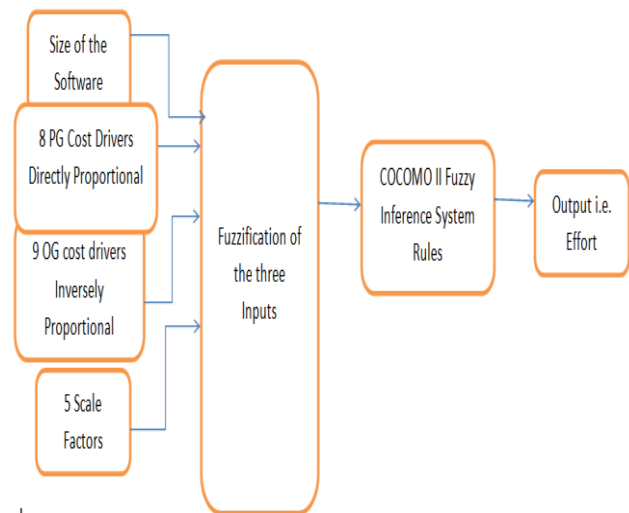

Figure: 1 Proposed Methodologies

We utilized Trapezoidal membership function, Gaussian membership function and Triangular membership function for analysis and the consequences of three are contrasted and the efforts processed by COCOMO II and the genuine aftereffects of the subset of NASA 93 task dataset from PROMISE software Engineering Repository data set which is publically accessible for research reason which comprise of 93 projects data from different focuses of different years. In fuzzification fuzzy rules are characterized by utilizing linguistic variables dependent on connective AND between Input variables.

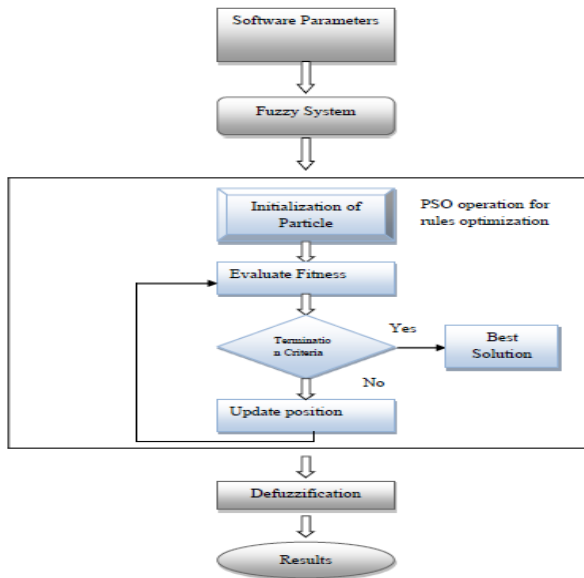The cost estimation in the proposed method is delineated in the Figure as a stream graph

Figure: 2"Proposed software cost estimation model"

Fuzzy triangular Membership function
The qualities having numerical values in the XML database is changed into the fuzzy utilizing the triangular membership function. Membership functions can either be picked by the client haphazardly or be structured utilizing AI methods like artificial neural networks, genetic algorithms and others. There are various states of membership functions; trapezoidal, triangular, Gaussian, piecewise-linear, bell-shaped, and so on. This investigation picked the Triangular membership function in which p, q and r speaks to the x directions of the three vertices of f(x) in a fuzzy set where, p is the inferior limit and r is the higher limit where membership degree is 0, q is where membership degree is 1. One of the key issues in every single fuzzy set is the means by which to choose fuzzy membership functions, The membership function completely characterizes the fuzzy set. A membership function gives a proportion of the degree of likeness of a component to a fuzzy set.
Membership functions can take any shape, however there are some steady models that show up in genuine applications.
The beneath Figure show a triangular membership function for a solitary fuzzy set. Here, it is comprehended that at p and r the worth is zero, and it comes to relentlessly to a limit of significant worth 1 at the middle point q between the p and r.
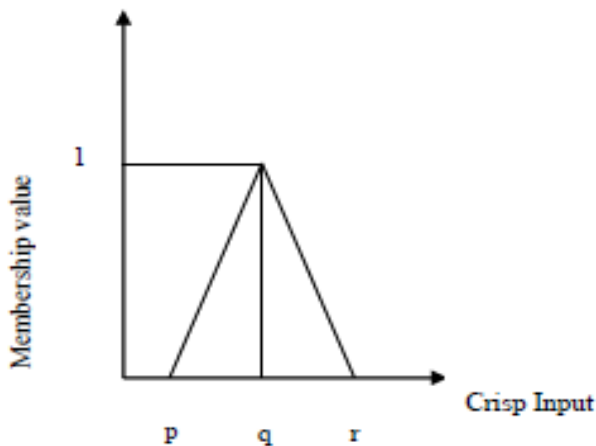

Figure:3 Triangular membership function
The beneath Figure shows the plot considering all the three membership functions having covering values. In this figure,

the bends for low, medium and high are appeared for the product efforts.
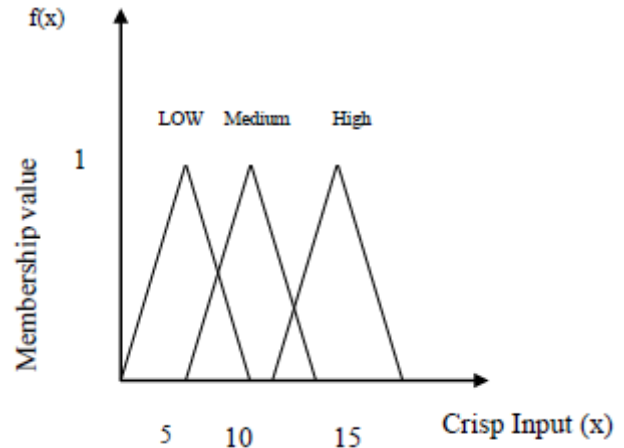

Figure: 4 Triangular membership functions with characterized parameters and their values

A. DEFUZZIFICATION

Preceding the defuzzification, conglomeration process is done in which fuzzy sets that speak to the outputs of each standard are joined into a solitary fuzzy set. The outcome of the total procedure is one fuzzy set for each output variable. The input for the defuzzification process is a fuzzy set (the total outcome fuzzy set) and the output is a solitary number. As much as fluffiness helps the standard assessment during the middle of the road steps, the last wanted output for every factor is commonly a solitary number. Be that as it may, the total of a fuzzy set envelops a scope of output values, and should be defuzzified so as to determine a solitary yield an incentive from the set. The Defuzzification method utilized is the weighted normal method. The cost evaluated through the proposed method is lesser when contrasted and different methods and it demonstrated to be a productive strategy in programming cost estimation. The trial examination of cost estimation is given beneath the entire criteria according to our specific situation.
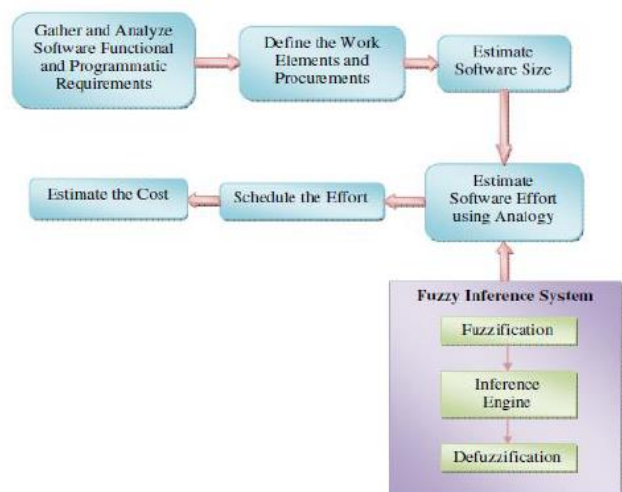

Figure: 5 Flow outline for the whole cost estimation process

IV. RESULT ANALYSIS

Programming cost estimation technique utilizing fuzzy logic with the guide of optimization algorithm is proposed as an improvement to the cost estimation technique utilizing delicate figuring. The proposed technique in assessed utilizing three distinct data sets. For each of these datasets the blunder measure like MRE and MMRE are calculated so as to discover the adequacy of the proposed framework. The datasets utilized in the investigation is the NASA 93 dataset. Different scientists have utilized distinctive blunder measurements. The most prominent mistake degree is Mean Absolute Relative Error (MARE).

$$MARE = \sum_{i=1}^{N} (|(E_i - A_i)/A_i|)/n$$

E_i - Estimated effort
A_i -Actual effort
The *MRE and* MMRE can be measured by employing the following formula

$$MRE = |E_i - A_i)|/A_i$$

where,
E_i - Estimated effort
A_i -Actual effort.
The MMRE for the estimated effort can be calculated by using the equation given below. The MMRE for the proposed method seems to be better than other works where fuzzy is used.

$$MMRE = \frac{1}{N} \sum_{i}^{N} MRE_i$$

where,
MRE - Mean relative error for i[th] measured error
n - Total number of measured errors.
Performance evaluation:
The Table 1 had given below shows the experimental results obtained in the proposed method. For various sizes the actual effort as well as the estimated efforts is calculated. The Magnitude of Relative Error for each entry is calculated. The actual effort has remained to be more reduced when compared with the estimated effort.

Table 1: Effort Estimates and MRE

No	Actual effort	Estimated Effort	MRE
1	68.5	67.251	0.0182336
2	147.1	143.13	0.0270224
3	75.25	74.165	0.0144186
4	133.17	129.25	0.0294506
5	102.22	101.23	0.0096849
6	97.23	89.12	0.0834105
7	47.125	48.12	0.0211114
8	65.23	59.16	0.0930553
9	112.35	110.25	0.0186915
10	46.25	41.22	0.1087567
11	125	121.5	0.0280000
12	129.2	123.52	0.0439814
13	72	76.677	0.0649556
14	83.25	79.23	0.0482883
15	132.25	112.25	0.1512287
16	60.23	56.013	0.0700166
17	51.21	48.078	0.0611619
18	354	333.64	0.9049850
19	392.54	314	0.2000815
20	45.25	39.56	0.1257458
21	278.2	254.2	0.0862688

22	184.2	169.2	0.0814332
23	150.23	130.57	0.130866
24	78	95.067	0.2188102
25	227	213.21	0.0607621
26	244.36	235.21	0.0374447
27	225	238.05	0.0580044
28	354	365.45	0.0323474
29	180	153.65	0.1464167
30	227	211.63	0.0677181

Figure 6 shows the graphical representation of the effort value obtained from the proposed method. The actual effort is considerably higher when compared to the estimated effort values
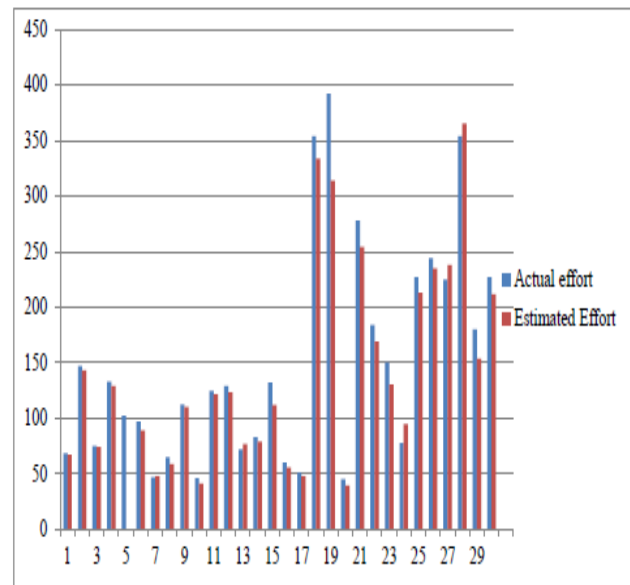

Figure 6 Graphical representations for Effort Estimates.

The proposed method is then contrasted and some existing method so as to assess the adequacy of the proposed technique. The MMRE measurements for the proposed and existing methods are given in Table 2. The MMRE measure is estimated in rate values.

Table 2 "MMRE measurements for the proposed and existing methods"

METHODS	MMRE (%)
Proposed Method	2.86
Fuzzy method	32.651

The graphical portrayal for examination of projected method with the existing method dependent on MRE and MMRE measures is appeared in the figure 7 beneath. Here, the MMRE measure for the dataset of the projected method is contrasted with the existing method. From the table and diagrams appeared, plainly the proposed method is increasingly effective when contrasted with the existing methods.
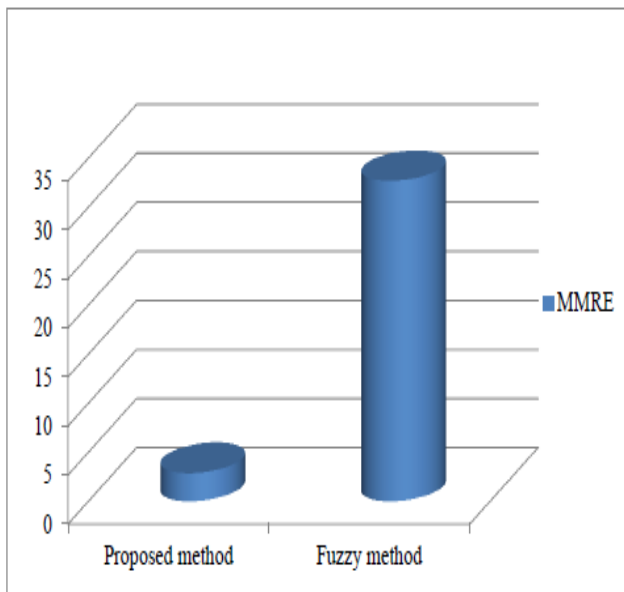
Figure 7 "The correlation of MMRE measure for the proposed method"

V. CONCLUSION

The examination has propelled an advanced plan to assess the product venture exertion, which is essentially founded on fuzzy logic and optimization process. In the fuzzy relationship model, both clear cut and numerical data are considered by fuzzy sets and subsequently, while delivering rules the investigation sends the optimization by methods for PSO with an eye on improving the principles for predominant functioning. The test results talked about demonstrate the predominance of this method when contrasted and existing techniques.

The performance analysis of proposed work for NASA93 dataset is clearly depicted above with various error measures. The study shows that the projected method yields greater results in estimating the project effort when compared with the existing method.

Early effort estimation is a vigorous task in earlier stages of project development. Success of any estimate model depends up on validity and reliability in several accuracy metrics. Future work is planned to study with other soft computing models and with large data sets testing.

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. B. W. Boehm, Software Engineering Economics, Englewoods Cliffs, NJ, Prentice-Hall, 1981.
3. Sweta Kumari, Shashank Pushkar Performance Analysis of the Software Cost Estimation Methods: A Review International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 7, July 2013.
4. O.Benediktsson and D. Dalcher, Effort Estimation in incremental Software Development, IEEE Proc. Software, Vol. 150, no. 6, pp. 351-357, December 2003.
5. R. K. Yadav and S. Niranjan, Project cost estimation using cocomo-2 metrics with fuzzy logic, Indian Journal of Science and Technology, vol. 10, no. 29, pp. 1-8, 2017.
6. T. A. Khalid and E. T. Yeoh, Early cost estimation of software reworks using fuzzy requirement-based model,- in International Conference on Communication, Control, Computing an Electronics Engineering. IEEE, pp. 1-5, 2017.
7. A. Kaushik, D. K. Tayal, K. Yadav, and A. Kaur, Integrating rey algorithm in artificial neural network models for accurate software cost predictions,- Journal of Software: Evolution and Process, vol. 28, no. 8, pp. 665-688, 2016.
8. Aron, J. D., "Estimating resources for large programming systems", NATO Science Committee, Rome, Italy, 1969.
9. Driankov D, Reinfrank M, Hellendoorn H. An Introduction to Fuzzy Control. Berlin: Germany. Springer; 1993.
10. Russo F. Fuzzy systems in instrumentation: Fuzzy signal processing. IEEE Trans Instrumentation and Measurement. vol.45. no.2.
11. Reddy, CS & Raju, K, 'Improving the accuracy of effort estimation through fuzzy set representation of size', Journal of Computer Science, vol. 5, no. 6, pp. 451, 2009.

AUTHORS PROFILE

Siva Suryanarayana Ch is completed MCA in 1996 from Osmania University, Hyderabad. M.Tech (Information Technology) in 2005 from AAI, Allahabad .I have guided more than 100 Graduate student's projects. Total teaching experience 24 years and 6 papers published in National and International journals.

Satya Prakash Singh is completed M. Sc(Physics) in 1997 from Dr B R Ambedkar University, Agra. M. Tech(Computer Science & Engineering) in 1999 from Kurukshetra University, Kurukshetra and Ph. D in 2012 from JJT University, Jhunjhunu, Rajasthan. Two Ph. D students completed Ph.D. under my supervision. I have guided more than 100 Post Graduate student's projects. Total teaching experience 20 years and more than 41 papers published in National and International journals.
.