Analogy based Software Effort Estimation and Accuracy Improvements

Rathi. J, Kamalraj. R, Karthik. S

Abstract— Accurate software development estimation in the software development is essential in the software project management practices and it will be affected completely by the irrelevant and the misleading information's. Impact of an excessive budget and project schedule concentrate becomes a rule in the today's IT industry. So that the software effort estimation is being a big challenge to software engineers for give effective management strategies. For software effort or cost estimation, analogy-based problem solving is extensively accepted by the software experts. Similarity measures play a vital role in the analogy based effort estimation. Distance between the project being estimated and the historical or the existing projects gives the final decision to generate effort estimation. The nearest neighbors of a project give the immutable conclusions for the present project. Here we taking the Desharnais data set as a sample data set and proceeding the Subset selection, Feature weighting, Discretization, Adaptation mechanism, Analogy selection, BRICH Clustering was applied in this system to give the performance accuracy and performance comparison with the Greedy Agglomerative Clustering(GAC), Neural Networks and the BRICH Clustering. Using the BRICH clustering gives better results to compare the other techniques.

Index Terms— Analogy, Discretization, Feature Weighting, Similarity, software Effort Estimation.

1 INTRODUCTION

The software project management is very crucial task in software project management especially for the software development effort estimation. Moreover the situations are huge deviation between the actual effort and an estimated effort. Hence the accurate effort estimate is highly desired. . Attributes of the effort estimation are Team Experience, Manager Experience, Year End, Length, Effort, Transactions, Entities, Points Adjust, Envergure, Points Non Adjust, Language. Effort estimation being a overestimation causes the loss of customers and the effort estimation being a underestimation cause loss of profit for company. Irrelevant and the misleading information's cause the wrong estimation. Find that confused situation for estimation and should remove those situations. Software Engineering is the technological managerial discipline concerned with systematic production and maintenance of software products that developed and modified on time and within cost estimates [26].

The primary goals of the software engineering are to complete the project within the budget and the time period. So that to meet best effort estimation strategies and experts to do. Software Effort Estimation is often occurring phenomenon in our every day lives. And the effort estimation parameters are the cost, resources, personnel and equipment. Some other limits like time, schedule and other similar attributes. A good estimation is the right mix of the parameters and their attributes. Software project success based on the good software effort estimation otherwise it cause the failure on it. And the ingredients of a good estimation are activity scope, Work environment, consistency, the usage of tools and learning strategies for the past experience. When develop software the scope is, the size of the software in terms of functionality and in terms of lines of code delivered. The environment in which the activity execute, impact on the overall estimation.

Rathi.J[1], II-ME (Software Engineering) SNS College of Technology,
E-Mail: rathinaidu.me@gmail.com, Coimbatore, TamilNadu.
Kamalraj.R[2], AP/CSE, SNS College of Technology,
E-Mail:mailtokamalraj@yahoo.com, Coimbatore, TamilNadu.
Karthik.S[3], Dean/CSE, SNS College of Technology,
E-Mail: profskarthik@gmail.com, Coimbatore, TamilNadu.

Estimation Approaches

The two major approaches in effort estimation are the heuristic approach and parametric approach.

Heuristic Approach

In this approach, the professionals experiment and find solutions from the frequently occurring problems. And it derived from the software experts, software gurus and experienced professionals developed and evaluated through repeated projects.

a. Expert-Based

When the quantified and empirical data are absent, in this situation this method is very useful to the estimators. Large community of software professionals believed their own past experience than using the estimation models developed by other professionals.

b. Analogy Method

In this method use the experience of the past projects. It compares the proposed project to previously completed and similar projects.

c. Bottom-Up Method

This method estimates the each component of the software project individually and combines all the results and it produces the final estimation value. Here we need to define each component and activity.

d. Top –Down Method

In this method refers the Work Breakdown Structure

(WBS). It working through the main modules, sub modules and individual functions.

e. Algorithmic Method

Experts observed some data patterns and based on it this method conceptualized. In this pattern transformed into mathematical formulae that used to derive the software estimations.

Parametric Approach

- Larry Putnam's SLIM (Software Life-Cycle IModel).
- Galorath's SEER-SIM based on the Jensen Model.
- Object Factories SELECT Estimator based on the Object Matrix model.
- Barry Boehm's COCOMO II based on the ingenious model.
- COSMIC's (Common Software Measurement International Consortium) COSMIC-FFP.
- Function Point by Allan Albrecht and later by International Function Point Users Group (IFPUG).
- Knowledge Plan from Software Productivity Research.

2. RELATED WORK

2. 1 Impact Of Irrelevant And Misleading Information On Software Effort Estimates

The effort estimation for software vigorously affected by the irrelevant and misleading information's. Five different software projects were taken for estimation in a various companies. The companies were allowed to do the original or the manipulated version of the software requirements. The manipulates like requirement information about requirements specification of reduced length and no change of content, restriction about low starting period and short development period. Need more effort to study the field about the project and the scope of the project. The model was proposed about the difference between the laboratory settings and field experiments.

2. 2 Data Mining Techniques For Software Effort Estimation

Expecting a predictive model need for the effort estimation for all the situations. But there is no univocal solution to solve it. All the effort estimation models suits for a particular strategy. It gives the benchmarking study beyond the effort estimation models. Benchmarking aspect these techniques are taken tree/rule-based models like M5 and CART, linear models-linear regression, nonlinear models (MARS, multilayered perception neural networks, radial basis function networks, and least squares support vector machines), and case-based reasoning. Experimental results shows the data mining techniques could make the worthwhile contribution for effort estimation techniques.

2.3 Reuse In Systems Engineering

In the IT environment the reusability is a major factor for software development. It can be segment of a project or a source code used again with or without modifications. The software effort estimation for reusability is the important task. If the code taken for reuse purpose it could not exceed the actual cost level. Constructive Systems Engineering Cost Model (COSYSMO) has the effort drivers like requirements, interfaces, algorithms, and operational scenarios. It provides the environment consist COSYSMO reuse extension, COSYSMO effort drivers, defining reuse weighs, practical validation of the COSYSMO.

2.4 Impact Of Budget And Schedule Pressure On Software Development

Exceeding the schedule and budget has head ache for the project managers. Studies behind the schedule are an art and must mix with the proper ingredients. Put the U-Shape curve and plot the plots the data as the budget and schedule.

Must achieve the budget and schedule pressure against the client and software development teams. Descriptive strategies are cyclic time, effort, process maturity, size, complexity, quality, budget pressure and schedule pressure. The software management and more theoretical studies need to find more effective negotiation strategies, deadline and budget setting policies.

2.5 Grey Relational Analysis With Genetic Algorithm For Software Effort Estimation

In software effort estimation implicit or subjective estimations are acceptable. GRA (Grey Relational Analysis) has introduced. Domain experts have done the estimation and produce the final results about the project. The grey can be combination of black and white. Black contains required information's not entirely available. And white contains required information's are entirely available. To build a formal software estimation need the grey relation analysis for solve the incomplete or uncertain effort drivers. It used for problem solving to the similarity measures between the complex relations. And the system uses the genetic algorithm; integrate with the grey relational analysis to perform the estimation well. This model scrutinizes the impact of integrating the grey relational analysis with genetic algorithm. Then the results were comparing with other software development estimation models and produced the results. And it shows the improved estimation accuracy among other estimation models. Grey relation model was depending on the historical projects. This process has applied on other software estimation domains like software size and software quality estimation models.

2.6 Analogy-Based Software Effort Estimation Based On Similarity Distances

Analogies mean that similarity between the pairs of the project. Similarity measures play a vital role in this concept. It

calculates the distance between the historical project and the knowledge database with the actual requirement. It results the most similar project to compare with that actual project requirements. This model also used the genetic algorithm and integration of grey relational analysis techniques. And the results compared with other estimation models where the estimation required. It contains the analogy based effort estimation based on the similarity measures between the pairs of the project.

2.7 An Intelligent Algorithm For Soft-Ware Effort Estimation

The main factor in the software effort estimation is accuracy of the project management strategies. In this approaches used the fuzzy logic and genetic algorithm play vital role to produce the accurate results. Cost and time plays important parameters among the client and the software development team. Accuracy parameters for software effort estimation are the memory used, search time, build time and error rate performance study. The genetic algorithm and fuzzy logic both perform intelligently for the effort estimation.

3 ANALOGY BASED EFFORT ESTIMATION

Analogy based effort estimations used for the test projects, need to find the similar projects has completed in that organization. Those types of projects called as training projects. Baseline Analogy Based Estimation is called as ABE0. In that ABE consist with a table format and each row contains one project. And the column contains the independent variables (features) of the present project and dependent variables (features) of the present project. The features like effort required to complete the appropriate phase or the module. The scaling measure is used for ensure the independent values also get the same degree of influence between the original and the training data set.

3.1 Subset Selection

The original data set is derived from knowledge data base or the training project. After the subset selection process has taking place. Sunset selection means that improving the set of training sets. It contains remove nothing, outlier and prototype mechanisms. Then remove nothing is not change of anything from the original dataset. It is used without change. Outlier method is the process of removing the large values those are not fit to the project features. So need to remove those situations. Prototype model contains the model to remove the unexpected situations as a dataset and gives the mitigation for very large or small data avail at the original data. In this project outlier method has take for the subset selection process.

3.2 Feature weighting Methods

It applicable for the independent features of the training projects. This technique used to remove the less informative independent projects. To remove the redundant and noisy features has multiplied by zero. Genetic Algorithm, WRAPPER, Analogy-X and Correlation-based method are the feature weighting schemas. Using the selected predictor variables the distance matrices were constructed. Correlate the two distances and show the distance function. Analogy-X used to get the feature weighting function for the proposed project.

3.3 Discretization

Discretization is process of partitioning continuous variables into categories.

- The range of a continuous attributes are divide into intervals.
- Categorical attributes are accepted by the classification algorithms.
- Data size has reduced by Discretization

Three types of attributes

- Nominal contains the values from an unordered set.
- Ordinal contains the values from an ordered set.
- Continuous consists of real numbers.

Discretization methods are

- Equal frequency where c_i ¼ c_j;
- Equal width, where b_{ip1} _ b_i is a constant;
- Entropy;
- PKID;
- Do nothing at all are the Discretization.

Entropy based Discretization

Entropy based Discretization based on the supervised discretization. Set of samples S, S1 and S2 are the partitions for the given sample S and the boundary T has taken, the entropy after partitioning is

$$E(S,T)=\frac{|S_1|}{|S|}Ent(S_1)+\frac{|S_2|}{|S|}Ent(S_2)$$

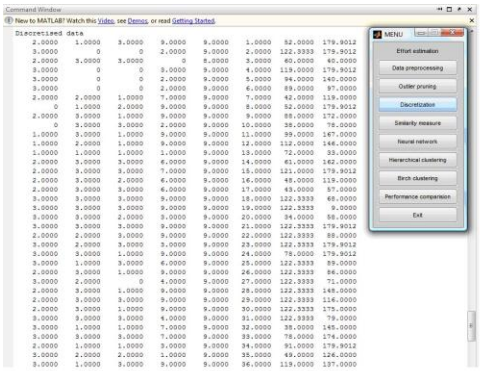


Fig: 1 shows the entropy discretization for the original dataset which is taken from the Desharnais data set.

3.4 Similarity Measures

A similarity measure is measuring the distance between two data objects and shows the closeness, results displayed into distance matrix. Identifying the similarity among all cases inside the data set. Methods of similarity measures are

- Jaccard distance for binary distance;
- Gower distance;
- Euclidean distance;

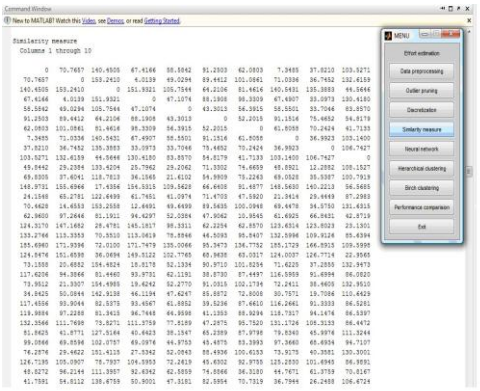


Fig: 2 Euclidean measures is the most fitting method for the software effort estimation. It generally based on the Pythagorean Theorem. Euclidean distance is used fort the analogy based effort estimation and the relevant features are software size, effort and duration of a project.

3.5 Adaptation Mechanism

The adaptation gives the decision to take the values to fit the present project features. It has some approaches for selecting it.

- Reporting the analogies median effort value.
- Reporting the mean-dependent value
- Regression, Model trees and Neural Networks for summarize the adaptations.
- Get the mean-dependent value.

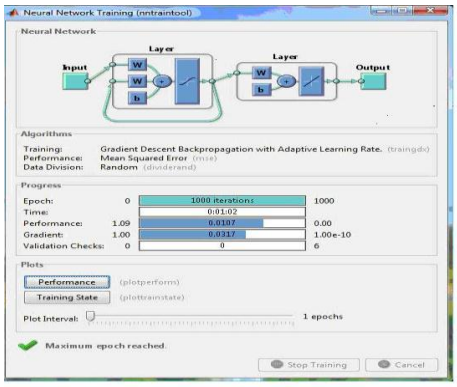


Fig : 3 in this project used neural networks for selecting the adaptation value. A neural network is a network of simulated neurons that can be used to recognize instances of patterns. A neuron fires when the sum of its collective inputs reaches a threshold. A simulated neural network node is connected to other nodes via links. Each link has an associated weight that determines the strength and nature (+/-) of one nodes influence on another. Influence = weight * output. Activation function can be a threshold function. Node output is then a 0 or 1. Real neurons do a lot more computation.

3.6 Selecting Analogies

The analogies are selected by two main methods. The fixed analogy selection method and dynamic analogy selection method. And fixed method used same set of analogies in the test set. Examples are k ¼ 1, k ¼ 2 and k ¼ 1,2,3. As per the task the analogies has set for the dynamic analogy selection. It used the Best (K) procedure.

- Randomly select N_T training projects.
- For each k 2 1 : : T_N, compute estimates for n 2 N.
- In step2 find the least error for k value.
- Use k- nearest neighbors and set k value.

3.7 Selection a Prediction Systems

The easy path limits the space of design options to just those that directly address the essential assumptions of the predictor. As shown below, for ABE this directs us to issues of case subset selection and the number of analogies used for estimation.

3.8 Identifying the Essential Assumptions

In the prediction system must identify the essential assumptions best fit for the project. In the analogy based estimation the similar projects that are more appropriate with the features of the historical and the present project requirements. Confuse estimation has calculated which is heavy different project values decrease the accuracy.

3.9 Identifying Assumption Violation

In this phase the violating situations are recognize in the essential assumptions. The k value has estimated and monitoring the larger k values and the smaller k values to identifying the violations. Hierarchal clustering algorithm has been used to generate the sub trees and the super tree. And the sub tree contains the nearest data values the super tree. The existing project the Greedy Agglomerative Clustering used to construct the tree. The hierarchical clustering used and it give the better results than the GAC clustering algorithm.

3.10 BRICH Clustering

Balanced Iterative Reducing and Clustering using Hierarchies (BRICH) is a hierarchical clustering type.

- Clustering decision is made without scanning all data points because BIRCH is local.

ISSN 2229-5518

- BIRCH accomplishment the observation that the data space is usually not commonly occupied, and hence not each data point is equally important for clustering purposes.
- BIRCH makes full use of available memory to derive the finest possible sub clusters (to ensure accuracy) while minimizing I/O costs (to ensure efficiency). Fig 4 shows the BRICH clustering results for the present project.
- Scans the database for build an in-memory tree in the given data set.
- Applies clustering algorithm among cluster the leaf nodes.

Fig 4 shows the BRICH Clustering result.

3.11 Performance Evaluation

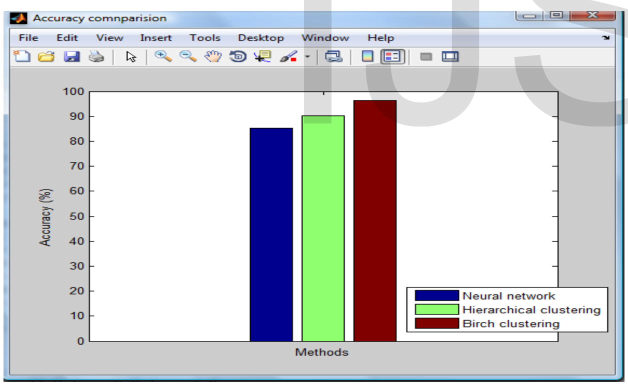


Fig: 5 for Accuracy Comparison

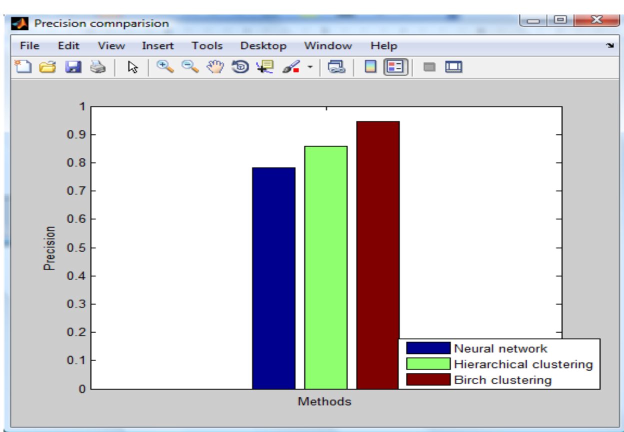
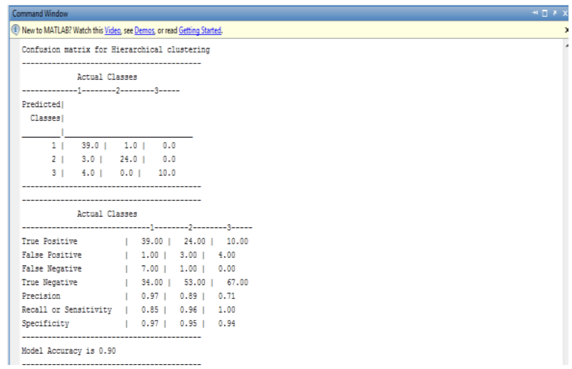


Fig : 6 for Precision Comparison

Fig 5 and Fig 6 shows the performance results between the neural networks, Hierarchical Clustering and BRICH Clustering. Compare the hierarchical clustering and neural networks, hierarchical cluster accuracy is high and comparing the BRICH Clustering and Hierarchical Clustering, BRICH Clustering accuracy is high.

3.12 Accuracy Results


Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Confusion matrix for Hierarchical clustering

Actual Classes

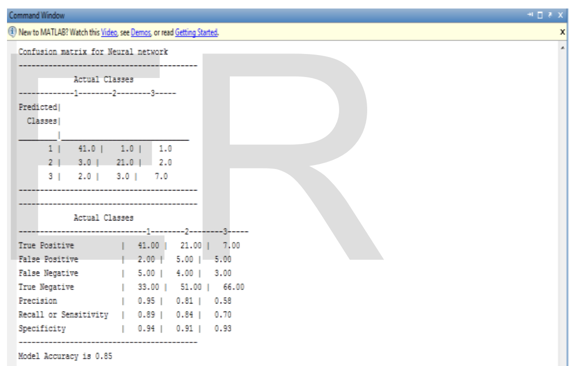
Predicted Classes	1	2	3
1	39.0	1.0	0.0
2	3.0	24.0	0.0
3	4.0	0.0	10.0

Actual Classes

	1	2	3
True Positive	39.00	24.00	10.00
False Positive	1.00	3.00	4.00
False Negative	7.00	1.00	0.00
True Negative	34.00	53.00	67.00
Precision	0.97	0.89	0.71
Recall or Sensitivity	0.85	0.96	1.00
Specificity	0.97	0.95	0.94

Model Accuracy is 0.90

Fig : 7 confusion Matrix for Neural Network


Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Confusion matrix for Neural network

Actual Classes

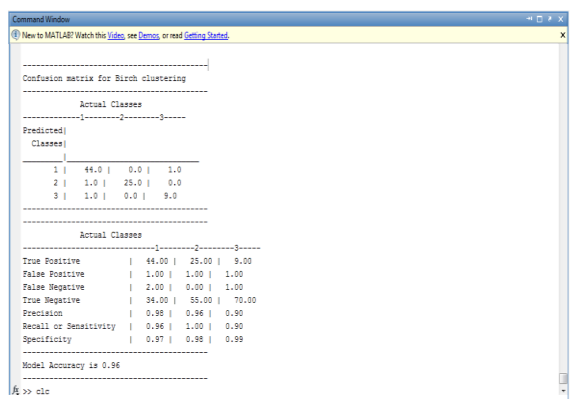
Predicted Classes	1	2	3
1	41.0	1.0	1.0
2	3.0	21.0	2.0
3	2.0	3.0	7.0

Actual Classes

	1	2	3
True Positive	41.00	21.00	7.00
False Positive	2.00	5.00	5.00
False Negative	5.00	4.00	3.00
True Negative	33.00	51.00	66.00
Precision	0.95	0.81	0.58
Recall or Sensitivity	0.89	0.84	0.70
Specificity	0.94	0.91	0.93

Model Accuracy is 0.85

Fig: 8 confusion Matrix for Hierarchical Clustering


Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Confusion matrix for Birch clustering

Actual Classes

Predicted Classes	1	2	3
1	44.0	0.0	1.0
2	1.0	25.0	0.0
3	1.0	0.0	9.0

Actual Classes

	1	2	3
True Positive	44.00	25.00	9.00
False Positive	1.00	1.00	1.00
False Negative	2.00	0.00	1.00
True Negative	34.00	55.00	70.00
Precision	0.98	0.96	0.90
Recall or Sensitivity	0.96	1.00	0.90
Specificity	0.97	0.98	0.99

Model Accuracy is 0.96

>> clc

Fig: 9 confusion Matrix for BRICH Clustering

Fig 7, Fig 8 and Fig 9 shows the accuracy rates and Fig 7 gives the result about the neural networks accuracy which is 85%. Fig 8 gives the hierarchical clustering accuracy which is 90%. Fig 9 shows the BRICH clustering accuracy which is 96%.

4 CONCLUSION AND FUTURE WORK

This paper has presented the various effort estimations and detail study about the analogy based effort estimation with data mining clustering (BRICH). And it gives the performance evaluation about the accuracy compared BRICH with neural networks and hierarchical clustering. Proposed BRICH Clustering gives better results than other clustering approaches. In future we can build other algorithms instead of proposed algorithm to obtain the better accuracy.

REFERENCES

1. Baker. D, "A Hybrid Approach to Expert and Model-Based Effort Estimation," master's thesis, LCSEE, West Virginia Univ., http:// bit.ly/hWDEfU, 2007.

2. Ekren Kocaguneli, Tim Menzies, Ayse Bener and Jacky W. Keung, "Exploiting the Essential Assumptions of Analogy-Based Effort Estimation", IEEE Trans on Software Engg, Vol. 38, No. 2, March/April 2012.

3. Gan Wang, Ricardo Valerdi and Jared Fortune, "Reuse in Systems Engineering" IEEE Systems Journal, Vol. 4, No. 3, September 2010.

4. Heejun Park and Seung Baek, "An empirical validation of a neural network model for software effort estimation", Elsevier Expert Systems with Applications 35 (2008) 929–93.

5. Jorgensen. M and Gruschke. T, "The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments," IEEE Trans. Software Eng., vol. 35, no. 3, pp. 368-383, May/June 2009.

6. Jorgensen. M and Shepperd. M, "A Systematic Review of Software Development Cost Estimation Studies," IEEE Trans. Software Eng., vol. 33, no. 1, pp. 33-53, Jan. 2007.

7. Kadoda. G, Cartwright. M, and Shepperd. M, "On Configuring a Case-Based Reasoning Software Project Prediction System," Proc. UK Case Based Reasoning Workshop, pp. 1-10, 2000.

8. Karel Dejaeger, Wouter Verbeke, David Martens and Bart Baesens, "Data Mining Techniques for Software Effort Estimation: A Comparative Study", IEEE Transactions On Software Engineering, Vol. 38, No. 2, March/April 2012.

9. Keung. J. W, Kitchenham. B. A, and Jeffery. D. R, "Analogy-x: Providing Statistical Inference to Analogy-Based Software Cost Estimation," IEEE Trans. Software Eng., vol. 34, no. 4, pp. 471-484, July/Aug. 2008.

10. Kemerer .C, "An Empirical Validation of Software Cost Estimation Models". ACM, vol. 30, pp. 416-429, May 1987.

11. Kirsopp. C, Shepperd. M, and Premraj. R, "Case and Feature Subset Selection in Case-Based Software Project Effort Prediction," Proc. Int'l Conf. Knowledge-Based Systems and Applied Artificial Intelligence, 2003.

12. Kitchenham. B, Mendes. E, and Travassos G. H, "Cross versus Within-Company Cost Estimation Studies: A Systematic Review," IEEE Trans. Software Eng., vol. 33, no. 5, pp. 316-329, May 2007.

13. Li. Y, Xie. M and Goh. T, "A Study of Project Selection and Feature Weighting for Analogy Based Software Cost Estimation," J. Systems and Software, vol. 82, pp. 241-252, 2009.

14. Magne Jørgensen and Stein Grimstad, "The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experiment" IEEE Trans on Software Engineering, Vol. 37, NO. 5, Sep/Oct 2011.

15. Magne Jørgensen and Tanja M. Gruschke, "The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments", IEEE Transactions On Software Engineering, Vol. 35, No. 3, May/June 2009.

16. Magne . Jørgensen and Tanja M. Gruschke, "The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments", IEEE Transactions On Software Engineering, Vol. 35, No. 3, May/June 2009.

17. Martin Auer, Adam Trendowicz, Bernhard Graser, Ernst Haunschmid, and Stefan Biffl , "Optimal Project Feature Weights in Analogy-Based Cost Estimation: Improvement and Limitations", Ieee Transactions On Software Engineering, Vol. 32, No. 2, February 2006.

18. Menzies. T, Jalali. O, Hihn. J, Baker. D, and Lum. K, "StableRankings for Different Effort Models," Automated Software Eng., vol. 17, no. 4, pp. 409-437, Dec. 2010.

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. Mendes. E, Watson. I. D, Triggs. C, Mosley. N, and Counsell. S, "A Comparative Study of Cost Estimation Models for Web Hypermedia Applications," Empirical Software Eng., vol. 8, no. 2, pp. 163- 196, 2003.

21. Ning Nan and Donald E. Harter, "Impact of Budget and Schedule Pressure on Software Development Cycle Time and Effort" IEE Transactions On Software Engineering, Vol. 35, No. 5, September/October 2009.

22. Nan-Hsing Chiu and Sun-Jen Huang, "The adjusted analogy-based software effort estimation based on similarity distances", Elesever July 2006.

23. Parthasarathi. M. A, "Practical Software Estimation", Infosys Pearson Edu.

24. Randy K. Smith, Joanne E. Hale and Allen S. Parrish, "An Empirical Study Using Task Assignment Patterns to Improve the Accuracy of Software Effort Estimation", -IEEE Transactions On Software Engineering, Vol. 27, No. 3, March 2001.

25. Rathi. J, Kamalraj. R , Karthik. S, "Survey on Effective Software Effort Estimation Techniques" International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 1, No:8, OCT 2012.

26. Richard Fairley, "Software Engineering Concepts", TATA McGraw-Hill Edition.

27. Sun-Jen Huang, Nan-Hsing Chiu and Li-Wei Chen, "Integration of the grey relational analysis with genetic algorithm for software effort estimation", Elsevier 15 August 2007.

28. Sun-Jen Huang and Nan-Hsing Chiu, "Optimization of analogy weights by genetic algorithm for software effort estimation", Elsevier, Feb 2006.

29. Shepperd. M, "Software Project Economics: A Roadmap," Proc. Future of Software Eng., pp. 304-315, 2007.

30. Vahid Khatibi and Dayang N. A. Jawawi, "Software Cost Estimation Methods: A Review", Journal of Emerging Trends in Computing and Information Sciences.Volume 2,no.1.

31. www. google.com

BIOGRAPHY

Rathi received M.Sc (Software Engineering) from Bannari Amman Institute of Technology affiliated to Bharathiar University. She was working as a lecturer in the Department of Computer application from Bannari Amman Institute of Technology, Sathyamangalam. And doing her final year Master of Engineering in (Software Engineering) in SNS College of Technology affiliated to Anna University Chennai. She is an active member in CSI Student Chapter. She has published a paper in international journal and attends an international conference. Her research includes Software Cost Estimation, Software Engineering, Software Testing, Data Mining and Cloud Computing.

R. Kamalraj received his B.E. degree in Computer Science &Engineering from Bharathiyar University, Coimbatore, Tamil Nadu, INDIA in 2002, the M.E . degree in Computer Science & Engineering from Anna University, Chennai, Tamil Nadu, INDIA in 2009, and pursuing Ph.D. degree in Software Testing and Quality Management at Anna niversity of Technology, Coimbatore. He has published 7 papers in international journals and 1 paper in National Journal. He is having 9 years of teaching experience in 4 different engineering colleges. At present he is working as an Assistant Professor in the Department of Computer Science and Engineering at SNS College of Technology, Coimbatore. His research interests include Software Testing, Software Quality Management and Data Mining.

Professor Dr.S.Karthik is presently Professor & Dean in the Department of Computer Science & Engineering, SNS College of Technology, affiliated to Anna University- Coimbatore, Tamilnadu, India. He received the M.E degree from the Anna University Chennai and Ph.D degree from Ann University of Technology, Coimbatore. His research interests include network security, web services and wireless systems. In particular, he is currently working in a research group developing new Internet security architectures and active defense systems against DDoS attacks. Dr.S.Karthik published more than 35 papers in refereed international journals and 25 papers in conferences and has been involved many international conferences as Technical Chair and tutorial presenter. He is an active member of IEEE, ISTE, IAENG, IACSIT and Indian Computer Society.