Performance Evaluation and Selection of Software Effort Estimation Models Based on Multi-Criteria Decision Making Method

Rakesh Garg

[1] Department of Computer Science & Engineering, Amity University, Uttar Pradesh (India)

Abstract: **Context: Software cost/effort estimation is a very critical job in the software development process that may be defined as the prediction process of the total effort required to develop any software. A number of software effort estimation models have been developed by the various researchers in the past but it is observed that none of them can be applied successfully in all kind of projects in different environments that raises the problem of the selection of the software effort estimation models. The selection of the software effort estimation models has been receiving more and more attention of the researchers now a day.**

Objective: The prime goal of the present research is to develop a methodology for the efficient and effective evaluation and selection of the software effort estimation models.

Methods/Approaches: In the present research, the software effort estimation model selection problem is represented as a multi-criteria decision making problem and a novel integrated approach namely Fuzzy-TOPSIS (F-TOPSIS) is further proposed to solve the present problem.

Result: As the result of the research carried out, various software effort estimation models are ranked according to their performance index values. The model having the maximum performance index is ranked at top i.e. number -1 and with minimum performance index is ranked at last position.

Keywords: Software effort estimation models, Evaluation indexes; Fuzzy-TOPSIS, MCDM;

I. INTRODUCTION

In today scenario, software has grown to be a very significant component for all kind of systems and Software effort estimation is of vital importance for the well balanced management of imminent software. The management in terms of software effort is necessary due to the speedy increase in the need of complex software systems at a large scale. In the past decades, a number of models have been developed by the various researchers to portray the software effort estimation. The prime motive of the software developers is to develop software with better quality by minimizing the required effort. So, it can be stated that the success of any software to some extent relies on the perfect effort estimation. The inaccurate estimation of required effort for the development of any software will effect adversely to both the customers and developers of the software because it can result in bad quality software, delay in the delivery time, misuse of software and contact cancellation etc.

Due to the aforementioned requirements, the researchers focus on an emerging research problem termed as selection of software effort estimation model/technique. A comprehensive systematic review of the past literature enforces that the problem of selection and ranking of software effort estimation models can be represented in the form of multi-criteria decision making problem. The MCDM problems are those problems in which a set of existing alternatives are evaluated on a pre-identified set of evaluation criteria and the information about that "which alternative is best" is obtained as the result.

The present research argues on the problem "evaluation and selection of software effort estimation models" by modeling it as a MCDM problem. Further, a set of evaluation criteria is proposed that can contribute in the evaluation process of various software effort estimation models. The rest of the paper is ordered as: section 2 included the literature review in support to the present problem and the proposed methodology, section 3 contains the illustrated example followed by methodology validation. The results and conclusion are analyzed in section 4 under major findings and discussions of the paper.

II. METHODS AND MATERIALS

An extensive literature survey into two parts was carried out in the context of the present problem of software effort estimation model selection. In the first part, the various models developed by the various researchers were studied with their associated advantages and disadvantages. It is inferred from that the software effort estimation models provided in past are mainly categorized into two categories namely (i) Algorithmic and (ii) Non-Algorithmic models [1-3].

In the second part, the various evaluation indexes that contribute in the evaluation process of the software effort estimation models were analyzed. Leung and Fan provide selection criteria as mean relative error (MRE), mean absolute relative error (MARE), balance relative error (BREbias) to solve the present problem [4]. Moløkken-Østvold provides mean relative error (MRE), balance relative error bias (BREbias), balance relative error (BRE) and accuracy (ACC) as the selection criteria for the evaluation of software effort estimation models [5]. Tim Menzies et al. proposed a methodology based on heuristic rejection rules named coseekmo to rank the various software effort estimation models by considering mean relative error (MRE), mean magnitude of relative error (MMRE) and prediction (PRE) as selection criteria [6].

Basha and Dhavachelvan proposed several evaluation metrics as mean relative error (MRE), mean magnitude of relative error (MMRE), prediction (PRE), root mean square (RMS), relative root mean square (RRMS) for the ranking of software effort estimation models [7]. In the contemporary work, Kaur et al. used some attributes as mean magnitude of relative error (MMRE), mean square error (MSE), root mean square error (RMSE) and root mean square error (RMSSE) as the selection criteria [8]. Sehra et al. proposed a model based on Fuzzy Analytic Hierarchy Process by accounting reliability (REL), mean magnitude of relative error (MMRE), Prediction (PRE) and uncertainty (UNC) as selection criteria [9]. Malathi and Sridhar proposed different selection criteria for the selection and raking of effort estimation models such as prediction (PRED), value accounted for (VAF), variance absolute relative error (VARE), mean absolute relative error (MARE), magnitude of relative error (MRE), root mean square error (RMSE) [10]. Wen et.al provide prediction (PRED), mean magnitude of relative error (MMRE) and median magnitude of relative error (MDMRE) for the purpose of selection and evaluation of models [11]. Noel et al. developed a methodology based on fuzzy logic for the comparison of two fuzzy logic models for software development effort estimation. Prediction (PRE), mean error relative (MER), (mean magnitude of error relative) MMER were used as selection criteria in this research [12]. Mittas and Angelis ranked various effort estimation models by using different criteria as mean absolute error (MAE), magnitude of relative error (MRE) and mean magnitude of relative error (MMRE) [13]. Preeth et.al used three selection criteria for the evaluation of effort estimation model such as magnitude of relative error (MRE), mean magnitude of relative error (MMRE) and prediction (PRED) [14]. Nayebi et.al used different criteria such as prediction (PRED), correlation coefficient (CORR) and Bayesian information correlation (BIC) for the selection of different models [15].

In the present research, a hybrid approach namely F-TOPSIS is proposed by integrating two approaches as Fuzzy Set Theory (FST) and Technique for Order Preference Similarity to Ideal Solution (TOPSIS) for the first time to solve the problem of evaluation and selection of software effort estimation models. FST is used to assign the priority weights to evaluation indexes and the performance ratings to the software effort estimation models. To determine the aggregate priority weights and the performance ratings, average fuzzy operator was used. TOPSIS is further applied to calculate the performance index for each software effort estimation model by considering the separation of each alternative from the positive and negative ideal solutions. Finally, the alternatives i.e. software effort estimation models are ranked on the basis of their performance index values. Firstly, the criterion rating matrix is formed by taking the priority weights and performance ratings of various software effort estimation models as shown below.

$$CR_{ij} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \\ c_1 & c_2 & \cdots & c_n \end{bmatrix} \quad (1)$$

Now, the normalized decision and the weighted normalized decision matrix are formed as-

$$ND_{ij} = \begin{bmatrix} N_{11} & N_{12} & \cdots & N_{1n} \\ N_{21} & N_{22} & \cdots & N_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ N_{n1} & N_{n2} & \cdots & N_{nn} \end{bmatrix} \quad (2)$$

$$W_{ij} = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1n} \\ W_{21} & W_{22} & \cdots & W_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ W_{n1} & W_{n2} & \cdots & W_{nn} \end{bmatrix} \quad (3)$$

Now, the positive ideal solution and negative ideal solutions are determined from the above weighted normalized decision matrix as given below.

$$PIS = \begin{bmatrix} P_1 \\ P_2 \\ \cdot \\ \cdot \\ P_n \end{bmatrix} \qquad (4)$$

Where PIS= { W1* , …, Wn*}, where Wj* = { max (Wij) if j ∈ J ; min (Wij) if j ∈ J'}.

$$NIS = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{bmatrix} \qquad (5)$$

Where NIS={W_1' , …, W_n' }, where W' = { min (W_{ij}) if j ∈ J ; max (W_{ij}) if j ∈ J'}.

Finally, the performance index/suitability index value is calculated using SI = PIS / (PIS +NIS), 0< SI < 1.

III. AN ILLUSTRATED EXAMPLE

In the present research, eleven software effort estimation models under the algorithmic model namely Bailey and Basili Model (M1), Far and Zagorski model (M2), Nelson model (M3), Boeing model (M4), Cocomo model (M5), Aron model (M6), Putnam model (M7), Doty model (M8), Wolverton model (M9), Walston and Felix model (M10) and Jensen model (M11) are evaluated against eleven evaluation metrics namely Magnitude of Relative Error (MRE), Root Mean Square (RMS), Prediction (PRED), Root Mean Square Error (RMSSE), Mean Absolute Relative Error (MARE), Variance Absolute Relative Error (VARE), Value Accounted For (VAF), Accuracy (AC), Reliability (REL), Uncertainty (UNC), and Mean Absolute Error (MAE). Due to lack of maturity in the available secondary data, primary data was collected through the well designed questionnaires. Firstly, a team of five experts (E1, E2, E3, E4, E5) was established deliberately from academia, laboratories as well as IT industries. All the experts have more than 15 years experience in the field of the software effort estimation/software development. Secondly, two questionnaires were designed to collect the data regarding the priority weights of the evaluation metrics and the performance ratings of the software effort estimation models. The experts were asked to provide this data on a 7-point fuzzy scale and the data so obtained in linguistic terms from the experts was converted into a single value i.e. crisp value by performing simple average fuzzy operations on it. To check the reliability and accuracy of the data collected from the experts, reliability test and the ANOVA test was also performed. The priority weights of the evaluation indexes and the performance ratings of the various models and are provided in table 1 and Appendix-1 respectively.

According to proposed methodology, the criteria rating matrix can be formed using eq (1). After the formation of criteria rating matrix, the normalized and weighted normalized decision matrices are formed by using eqs. (2) and (3). Now, the

TABLE 1

PRIORITY WEIGHTS OF THE EVALUATION INDEXES

Evaluation indexes	Weights	Evaluation indexes	Weights
MRE	0.1323	VAF	0.0821
RMS	0.1198	AC	0.0696
PRED	0.1267	REL	0.0752
RMSSE	0.1030	UNC	0.0585
MARE	0.0988	MAE	0.0417
VARE	0.0919		

positive and negative ideal solutions are determined. In the next step, the separation of each software effort estimation model from the positive (0.083, 0.078, 0.079, 0.079, 0.055, 0.082, 0.060, 0.086, 0.060, 0.065, 0.079) and negative (0.042, 0.053, 0.040, 0.047, 0.075, 0.038, 0.063, 0.036, 0.065, 0.062, 0.041) ideal solution are determined that are used to calculate the performance index of each model to rank them. The rankings of all 11 models so obtained with their performance index values are given in table 2.

To show the applicability and utility of the proposed methodology i.e. F-TOPSIS, methodology validation is also carried out by making comparison of the obtained rankings results a well known MCDM approach namely Analytical Hierarchy process (AHP) developed as shown in table 3.

IV. MAJOR FINDINGS AND DISCUSSIONS

TABLE 2

RANKINGS OF SOFTWARE EFFORT ESTIMATION MODELS USING F-TOPSIS

Software Effort Estimation Models	Performance Index	Rank
M1	0.3375	8
M2	0.4049	5
M3	0.3372	9
M4	0.3723	6
M5	0.5765	1
M6	0.3147	10
M7	0.5153	3
M8	0.2944	11
M9	0.5185	2
M10	0.4887	4
M11	0.3432	7

According to the anticipated methodology, the better value of the performance index leads to the better ranking. The rankings of 11 software effort estimation models are represented in fig. 1. that shows Cocomo Model (M5) is ranked at number-1 i.e. at first position due to having maximum value of performance index (0.5765) in comparison to other models followed by Wolverton model (M9) at number-2 whereas the model namely Doty model

TABLE 3
COMPARATIVE RANKINGS OF SOFTWARE EFFORT ESTIMATION MODELS USING F-TOPSIS AND AHP

Software Effort Estimation Models	Rankings Obtained	
	AHP	F-TOPSIS
M1	10	8
M2	6	5
M3	8	9
M4	3	6
M5	4	1
M6	9	10
M7	2	3
M8	11	11
M9	1	2
M10	5	4
M11	7	7

(M8) is ranked at last i.e. number-11 having minimum performance index value (0.2944). Further, an attempt is made to validate the F-TOPSIS; a comparison with AHP is also performed in the research and shown graphically in fig. 2. It depicts some differences between the rankings of two methodologies as F-TOPSIS and AHP. These ranking differences occur because the evaluation indexes priority weights are not properly considered in AHP.
The main focus of the present research is to develop a methodology that can solve the software effort estimation model selection problem in an efficient and effective manner. The present problem is modeled as a multi-criteria decision making problem.

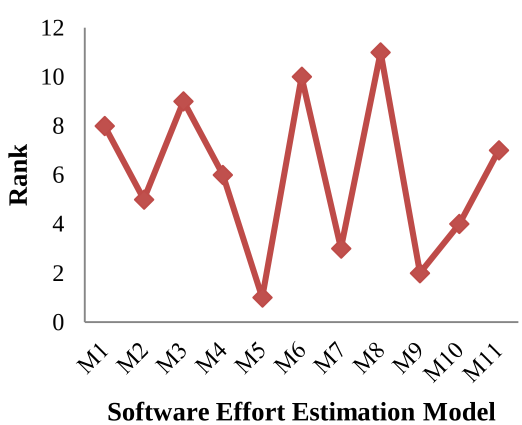

Fig. 1. Rankings of Software Effort Estimation Models obtained from F-TOPSIS

The alternatives i.e. software effort estimation models are

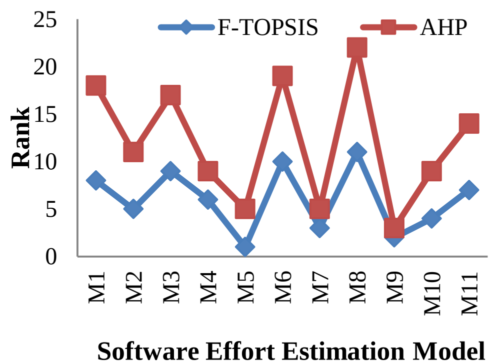

Fig. 2. Comparative Rankings of Software Effort Estimation Models obtained from F-TOPSIS and AHP

evaluated on a number of conflicting evaluation metrics and finally ranked

according to a single numeric value termed as performance index. The proposed methodology simple involves the use or mathematical matrix operations that make it simpler to implement. Further, the present research can be enhanced by performing comparisons with other MCDM approaches, sensitivity analysis and development of a computerized system.

References

[1] Abbas, S. A., Liao, X., Rehman, A. U., Azam, A., & Abdullah, M. I. (2012). Cost estimation: A survey of well-known historic cost estimation techniques.Journal of Emerging Trends in Computing and Information Sciences, 3(4), 612-63.
[2] Pandey, P. (2013, April). Analysis of the Techniques for Software Cost Estimation. In Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on (pp. 16-19). IEEE.
[3] Ramesh, K., & Karunanidhi, P. (2013). Literature survey on algorithmic and non-algorithmic models for software development effort estimation. International Journal of Engineering And Computer Science ISSN, 2319-7242.
[4] Leung, H., & Fan, Z. (2002). Software cost estimation. Handbook of Software Engineering, Hong Kong Polytechnic University, 1-14.
[5] Moløkken-Østvold, K., Jørgensen, M., Tanilkan, S. S., Gallis, H., Lien, A. C., & Hove, S. W. (2004, September). A survey on software estimation in the Norwegian industry. In Software Metrics, 2004. Proceedings. 10th International Symposium on (pp. 208-219). IEEE.
[6] Menzies, T., Chen, Z., Hihn, J., & Lum, K. (2006). Selecting best practices for effort estimation. IEEE Transactions on Software Engineering, 32(11), 883-895.
[7] Basha, S., & Ponnurangam, D. (2010). Analysis of empirical software effort estimation models. arXiv preprint arXiv:1004.1239.
[8] Kaur, J., Singh, S., Kahlon, K. S., & Bassi, P. (2010). Neural network-a novel technique for software effort

estimation. International Journal of Computer Theory and Engineering, 2(1), 17.
[9] Sehra, S. K., Brar, D., Singh, Y., & Kaur, D. (2013). Multi criteria decision making approach for selecting effort estimation model. arXiv preprint arXiv:1310.5220.
[10] Malathi, S., & Sridhar, S. (2012). Analysis of Size Metrics and Effort Performance Criterion In Software Cost Estimation. 2012.
[11] Wen, J., Li, S., Lin, Z., Hu, Y., & Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. Information and Software Technology, 54(1), 41-59.
[12] Garcia-Diaz, N., Lopez-Martin, C., & Chavoya, A. (2013). A comparative study of two fuzzy logic models for software development effort estimation. Procedia Technology, 7, 305-314.
[13] Mittas, N., & Angelis, L. (2013). Ranking and clustering software cost estimation models through a multiple comparisons algorithm. IEEE Transactions on Software Engineering, 39(4), 537-551.
[14] Preeth, R., ShivaKumar, N., Balaji, N.(2014). Software effort estimation using attribute refinement based adaptive Neuro Fuzzy Model. . International Journal of Innovative Research in Science Engineering and Technology 3(3).
[15] Nayebi, F., Abran, A., & Desharnais, J. M. (2015). Automated selection of a software effort estimation model based on accuracy and uncertainty.Artificial Intelligence Research, 4(2), p45.

Appendix-1

Performance Ratings of the Software Effort Estimation Models

Models/Evaluation indexes	MRE	RMS	PRED	RMSSE	MARE	VARE	VAF	AC	REL	UNC	MAE
M1	0.56	0.34	0.86	0.36	0.92	0.16	0.1	0.7	0.5	0.58	0.66
M2	0.78	0.11	0.66	0.26	0.96	0.54	0.28	0.7	0.62	0.74	0.92
M3	0.9	0.3	0.7	0.46	0.96	0.1	0.22	0.28	0.84	0.74	0.66
M4	0.9	0.34	0.88	0.36	0.99	0.1	0.22	0.28	0.97	0.86	0.8
M5	0.78	0.26	0.46	0.34	0.9	0.5	0.99	0.38	0.66	0.8	0.98
M6	0.56	0.16	0.86	0.3	0.92	0.22	0.28	0.74	0.5	0.58	0.8
M7	0.56	0.26	0.86	0.99	0.92	0.36	0.34	0.82	0.5	0.58	0.42
M8	0.74	0.14	0.7	0.5	0.42	0.1	0.22	0.89	0.5	0.92	0.78
M9	0.9	0.5	0.88	0.62	0.99	0.28	0.26	0.36	0.97	0.86	0.74
M10	0.56	0.46	0.86	0.89	0.92	0.28	0.26	0.16	0.5	0.58	0.66
M11	0.56	0.3	0.9	0.46	0.92	0.1	0.22	0.38	0.96	0.88	0.74