

# A Comparative Study of Estimation by Analogy using Data Mining Techniques

Geeta Nagpal\*, Moin Uddin\*\* and Arvinder Kaur\*\*\*

**Abstract**—Software Estimations provide an inclusive set of directives for software project developers, project managers, and the management in order to produce more realistic estimates based on deficient, uncertain, and noisy data. A range of estimation models are being explored in the industry, as well as in academia, for research purposes but choosing the best model is quite intricate. Estimation by Analogy (EbA) is a form of case based reasoning, which uses fuzzy logic, grey system theory or machine-learning techniques, etc. for optimization. This research compares the estimation accuracy of some conventional data mining models with a hybrid model. Different data mining models are under consideration, including linear regression models like the ordinary least square and ridge regression, and nonlinear models like neural networks, support vector machines, and multivariate adaptive regression splines, etc. A precise and comprehensible predictive model based on the integration of GRA and regression has been introduced and compared. Empirical results have shown that regression when used with GRA gives outstanding results; indicating that the methodology has great potential and can be used as a candidate approach for software effort estimation.

**Keywords**—Software Estimations, Estimation by Analogy, Grey Relational Analysis, Robust Regression, Data Mining Techniques

## 1. INTRODUCTION

Software development is a creative process where each person's efficiency is different. It is difficult to plan and estimate at the beginning as most software projects have deficient information and vague associations amongst effort drivers and the required effort. Software developers and researchers are using different methods and are more concerned about accurately predicting the effort of the software product being developed. Even a small enhancement in the prediction accuracy and validity are highly valued by researchers and software developers. During the last couple of years, research on software effort estimation has drifted from formal methods like Cocomo, Cocomo II, Function point and SLIM to Estimation by Analogy (EbA). In EbA, the effort required to develop a new software project requires the gathering of the particulars of the desired project. The particulars are then compared to a past project from the dataset having approximately the same specifications, but the challenge is the inconsistency in the struc-

---

※ The authors would like to thank the Promise Repository for providing such valuable datasets for our being able to carry out our research work

Manuscript received April 20, 2012; first revision July 4, 2012; accepted October 8, 2012.

**Corresponding Author: Geeta Nagpal**

\* Dept. of Computer Science and Engineering, National Institute of Technology, Jalandhar (sikkag@gmail.com)

\*\* Delhi Technological University, Delhi, India (prof\_moin@yahoo.com)

\*\*\* University School of IT, Gurugobind Singh Indraprastha University, Delhi, India (arvinderkaurtakkar@yahoo.com)

ture of the datasets used and in the feature set classification. The motivation behind *EbA* is that comparable software projects have similar costs.

*EbA* methods range from machine learning methods, regression techniques, the Grey System Theory (GST), and soft computing methods to a combination of these. GST is a recently developed system engineering theory that was first established by Deng in 1982 [1-3]. It draws out valuable information by generating and developing the partially known information. It has been applied in different areas of image processing [4], mobile communication [5], machine vision inspection [6], decision making [7], stock price prediction [8], and system control [9]. Some of the research carried out using *EbA* has been discussed. Mukhopadhyay et al. [10] developed ESTOR, which is a CBR tool to estimate project effort. The metrics used by ESTOR are the function point components and inputs to the intermediate COCOMO model. Shepperd et al. [11], developed ANGEL, which is a tool that supports the collection, storage, and identification of the most analogous projects, in order to estimate the effort for a new project. ANGEL uses Euclidean distance as the similarity metric in  $n$  dimensional space. Angelis et al. [12] uses a statistical simulation modus operandi to improve estimation by analogy. They investigated the problem of determining the most favourable method parameter configuration before application. The ANALOGY-X [13] tool provides a sound statistical basis for analogy based estimation using the Mantel's correlation randomization test. It removes the need for heuristic search, and greatly improves the algorithmic performance of the model. Baskeles et al., in [14] propose a machine-learning based model for software effort estimation and tested it on three different data sets, namely, NASA, USC, and SoftLab Data Repository (SDR). Idri et al. [15] suggested an approach based on reasoning by analogy, fuzzy sets, and linguistic quantifiers for software effort estimation when the project is described by either categorical or numerical data. Azzeh et al. in [16] integrated an analogy-based estimation with fuzzy numbers to improve the performance of software project effort estimation during its early development stages, using all of the available early data. They proposed a new similarity measure technique based on fuzzy numbers. Shepperd et al. [17] proposed a Grey Relational Analysis based software project effort (GRACE) prediction method, including a feature subset selection. Hsu and Huang [18] integrated six different weighted measures with the GRA method, namely, non-weight, distance-based weight, correlative weight, linear weight, nonlinear weight, and maximal weight, were applied to 127 projects from the ISBSG data set. GRACE<sup>+</sup>[19] addresses some of the theoretical challenges in applying GRA to outlier detection, feature subset selection, and effort prediction. In [20], Chiu et al. used genetic algorithms to optimize the analogy weights for software effort estimation. In [20], they integrated GA with GRA. Kosti et al.[21] suggested a new algorithm that initiates the notion of distance. It uses the similarity between distributions of distances instead of leaving one out cross validation to find the number of appropriate neighbours to be used for estimating the effort for new projects. AQUA [22] combines ideas from two known analogy-based estimation techniques: case-based reasoning and collaborative filtering. The method is applicable to predict efforts related to any object in the requirements, features, or project levels. In this research study, GRA, which is a technique of GST, utilizes the concept of absolute point-to-point distance between cases [17]. GRA is used to find the number of comparative projects ( $k$ ) that are closest to the reference project from the total of  $n$  projects. The projects are ranked based on their Grey Relational Grade's (GRG). Regression techniques namely, the Ordinary Least Square (OLS), Robust Regression (RR) techniques and Stepwise Regression (SWR) are applied to the  $k$  most influential projects in order to estimate the effort of the reference project. The value of  $k$  varies

with each reference project. The results obtained have shown an improvement over the conventional data mining techniques and also a significant improvement over GRACE [17], GRACE<sup>+</sup> [19], and FGRA [23]. A brief literature overview of the various analogy based approaches for software effort estimation is provided in Table 1. The remainder of the paper is organized as follows: Section II is a brief review of GRA and various regression techniques. Section III describes the proposed software effort prediction mechanisms using *GREAT\_RM*, the validation and evaluation criteria, and the datasets used in the methodology. Section IV provides the experimental results of *GREAT\_RM* and its comparison with some of the conventional data mining methods for software cost estimation along with their statistical significance. Section V provides, comparison of the results with some previously published results. Section VI presents threats to validity. The conclusion and directions for future work are presented in Section VII.

Table 1. Literature Overview of the Application of Analogy Based Approaches for Software Effort Estimation

Sr No.	Author/Year	Title	Journal/Conference	Approach/Technique	Datasets	Performance						Testing Mode	
							MRE	R <sup>2</sup>	Eqn				
1.	Srinivasan K, Fisher D, (1995)[24]	Machine Learning Approaches to Estimating Software Development Efforts	<i>IEEE Transactions on Software Engineering</i>	ANN, CART	Kemerer, COCOMO-81	CartX	364	0.83	102.5+0.075x		Holdout		
						B.Prpgt	70	0.80	78.13+0.88x				
						Func pt	10.3	0.58	-37+0.96x				
					Cocomo	610	0.70	27.7+0.156x					
					Slim	772	0.89	49.9+0.082x					
2.	Shepperd N, Schofield C, (1997)[11]	Estimating Software Project Efforts Using Analogies	<i>IEEE Transactions on Software Engineering</i>	OLS, Case based reasoning	Desharnais Finnish Kemerer Mermaid Telecom	Dataset	MMRE (%)	Reg1	Reg2	Pred (25%)	Reg1	Reg2	Cross Validation
						Desharnais 1	37	41	41	47	45	45	
						Desharnais 2	29	29	29	47	48	48	
						Desharnais 3	26	36	49	70	30	50	
						Mermaid E	53	62	62	39	27	27	
Mermaid N	60	--	--	25	--	--							
3.	Witting G, Finnie G, (1997)[25]	Estimating Software Development Efforts with Connectionist Models	<i>Information and Software Technology</i>	ANN	Simulated data Desharnais	ARE	%		Cum %		Holdout		
						0-10%	40.0		40.0				
						11-25%	36.7		76.7				
						26-50%	20.0		96.7				
						>50%	3.3		100				
4.	Burgess C.J., Lefley M., (2001)[26]	Can Genetic Programming Improve Software Effort Estimation? A Comparative Evaluation	<i>Information and Software Technology</i>	Genetic Algorithm	Desharnais	Correlation	0.752					Random	
						AMSE	11.13						
						Pred(25)	4.2						
						Pred(25)%	23.3						
						MMRE	44.55						
						BMMRE	75						
5.	Essam <i>et al.</i> , (2002)[27]	Software Project Effort Estimation Using Genetic Programming	<i>IEEE</i>	Genetic Algorithm	ISBSG	MSE	Tr = 2.90 Ts = 5.4					Random	
						R <sup>2</sup>	Tr = 0.44 Ts = 0.40						
						MMRE	Tr = 2.67 Ts = 1.91						
						Pred(25%)	Tr = 0.19 Ts = 0.21						
						Pred(50%)	Tr = 0.39 Ts = 0.40						
						MMRE	22.5						
						Pred(25%)	62.14						
6.	Idri <i>et al.</i> , (2002)[28]	Estimating Software Project Efforts by Analogy Based on Linguistic Values	<i>Eighth IEEE International Symposium on Software Metrics</i>	Fuzzy Analogy		MMRE	22.5					-	
						Pred(25%)	62.14						

Table 1. <Continue>

Sr No.	Author/Year	Title	Journal/Conference	Approach/Technique	Datasets	Performance				Testing Mode				
						Pred (20%)		Pred (30%)						
7	Huang X, Caretz L.F., and Ren J., (2003)[29]	A Neuro-Fuzzy Model for Software Cost Estimation	<i>Third International Conference on Quality Software, (QSIC'03)</i>	Neuro-Fuzzy, COCOMO II	COCOMO'81+ 6 industry projects.	86		92		All				
						Pred (20%)		Pred (30%)						
8	Song Q., Shepperd M., and Carolyn M, (2005)[17]	Using Grey Relational Analysis to Predict Software Efforts with Small Data Sets	<i>IEEE International Software Metrics Symposium</i>	Analogy, Grey Relational Analysis	Albrecht, NASA, COCOMO81, Desharnais, Kemerer	MMRE	MdMRE	Pred (25%)	Bias	Jack Knifing				
						60.25	21.35	52.63	-12.13					
						32.88	28.38	46.67	17.34					
						76.09	60.52	20.63	-18.89					
						49.83	33.93	30.00	-16.52					
58.83	46.94	26.67	-7.07											
9	Bohem et. al., (2005)[30]	Feature Subset Selection Can Improve Software Cost Estimation Accuracy	<i>PROMISE'05</i>	Feature Subset Selection (WRAPPER), COCOMO	COCOMO'81, NASA60, Project04, Project03, Project02	Mean (Pred(30))		Median(Pred(30))		Random				
						51.3		7.66						
						81.3		7.32						
						66.7		6.92						
						55.8		15.65						
10	Sentas P, Angelis L, Stamelos I, (2005)[31]	Software Productivity and Effort Prediction with Ordinal Regression	<i>Information and Software Technology</i>	OLS regression, Ordinal regression	COCOMO-81, Maxwell, ISBSG R7	Fitting accuracy for all 52 projects	MMRE	PRED 20	PRED 25	Holdout				
						33.28%	44.2%	53.8%						
						Fitting accuracy for learning dataset	34.38%	40.5%	50.0%					
						Predictive accuracy for the test data set	45.06%	30.0%	40.0%					
11	Sheta A F, A.(2006)[32]	Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects	<i>Journal of Computer Science</i>	Genetic Algorithm, COCOMO	NASA	VAF		96.31		Random				
12	Auer M, Trendowicz A, Biffl S, Haunschmid E, Graser B (2006)[33]	Optimal Project Feature Weights in Analogy-Based Cost Estimation: Improvement and Limitations	<i>IEEE Transactions on Software Engineering</i>	Case base reasoning	ESA, Desharnais, DPS database, Kemerer, Albrecht	Dataset	MMRE	Pred 25	VAR <sub>r</sub>	MFVW	Cross Validation			
						Albrecht	15	20	21	34				
						Desharnais 44	12	11	20	43				
						Desharnais 23	13	19	23	65				
						Desharnais 10	21	20	30	18				
						ESA 29	8	7	1	58				
						ESA 13	3	4	9	29				
						Kemerer 15	2	16	3	34				
						Laturi 12	6	14	3	8				
Laturi 11	16	25	20	27										
13	Baskales B Turhan B Bener A, (2007)[14]	Software Effort Estimation Using Machine Learning Models	<i>22<sup>nd</sup> International Symposium on Computer and Information Science</i>	RBF, MLP, SVM, Decision Tree		MMRE		Pred(30)		-				
						22.11		73.25						
						110.94		33						
						25.72		75						
						12.85		83.75						
14	Chiu N, Huang S (2007)[34]	The Adjusted Analogy-Based Software Effort Estimation Based on Similarity Distances	<i>The Journal of Systems and Software</i>	OLS regression, artificial neural network, CART, case based reasoning	DPS database, Abran	Model		MMRE	MDMRE	PRED 0.25		Cross Validation		
						Euc Dist	AE	train	test	train	test		train	test
							AAE	0.96	1.25	0.51	0.58		0.33	0.19
							Imp %	0.60	0.52	0.20	0.36		0.60	0.46
						Man Dis	AMH	38	58	61	38		82	126
							AAMH	1.03	1.01	0.59	0.51		0.24	0.29
							IMP %	0.66	0.49	0.26	0.31		0.52	0.38
						Minkowski Dis	AMK	36	52	56	39		117	31
							AAMK	0.99	1.24	0.55	0.55		0.31	0.24
							IMP %	0.61	0.49	0.29	0.38		0.50	0.38
						Mean imp %		37	57	55	36		87	72

Table 1. <Continue>

Sr No.	Author/Year	Title	Journal/Conference	Approach/Technique	Datasets	Performance				Testing Mode			
15	Keung J.W, Kitchenham A and Jeffery, D R (2008)[13]	Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation	IEEE Transaction on Software Engineering	Analogy, Jackknife	Desharnais	Mental* R		0.440		All			
						UCL		0.466					
16	Huang S J, Chiu N H and Chen L W, (2008)[20]	Integration of Grey Relational Analysis with Genetic Algorithm for Software Effort Estimation	European Journal of Operational Research, Elsevier	Grey Relational, Genetic Algorithm	COCOMO	MMRE		Pred(0.25)		Random			
						Tr	0.53	0.41					
						Ts	20.69	0.38					
						Albrecht	0.37	0.49					
					Ts	0.31	0.48						
17	Kiran et al., (2008)	Software Development Cost Estimation Using Wavelet Neural Networks	The Journal of Systems and Software	Wavelet Neural Network	Albrecht	MMRE		Pred(0.25)		Random			
						0.121	0.708		0.177				
						Desharnais	0.666		0.163				
18	Azzeh M, Neagu D, and Cowling P, (2008)[36]	Improving Analogy Software Effort Estimation Using the Fuzzy Feature Subset Selection Algorithm	PROMISE'08	Feature Subset Selection, Fuzzy Algorithm	Desharnais	MMRE		MdMRE		Pred(25)		Jack Knifing	
						28.7	21.8		54.7				
						40.2	32.4		39.8				
19	Azzeh M, Neagu D and Cowling P I, (2010)[23]	Fuzzy Grey Relational Analysis for Software Effort Estimation	Empir Software Eng, Springer	Analogy, Fuzzy Grey Relational	COCOMO'81	MMRE		MdMRE		Pred(0.25)		Jack Knifing	
						33.3	22.0	28.6		55.2			
						Deshrnais	30.6	17.5	34.4		64.7		
						Kemerer	36.2	33.2	34.3		52.9		
						Albrecht	51.1	48.0	60.4		28.6		
20	Azzeh M, Neagu D and Cowling P I, (2010)[16]	Analogy-Based Software Effort Estimation Using Fuzzy Numbers	The Journal of Systems and Software	Analogy, Fuzzy Numbers	COCOMO'81	MMRE		MdMRE		Pred(25)		Jack Knifing	
						28.55	17.80		59.80				
						33.37	20.36		62.33				
						Desharnais	26.89	19.32		64.94			
						Albrecht	50.08	30.75		50.00			
					Kemerer	55.65	24.24		53.33				
21	Chaudhary K, (2010)[35]	GA Based Optimization of Software Development Effort Estimation	IJCSI	SEL model, Walston-Felix model, COCOMO, Genetic Algorithm	Effort				Random				
22	Hari et. al. (2011)[37]	CPN-A Hybrid Model for Software Cost Estimation	IEEE Explorer	COCOMO, PSO, Neural Network	COCOMO'81	MARE	Tr = 19.49				Random		
							Ts = 10.96						
23	Song and Shepperd et al., (2011)[19]	Predicting Software Project Effort: A Grey Relational Analysis Based Method	Expert System with Applications	Analogy, Fuzzy Numbers	Albrecht	MMRE		Pred(25)		-			
						26.1	50						
						COCO NASA	24.2		60.3				
						COCOMO'81	49.8		29				
						Desharnais	41.4		45.3				
					Kemerer	19.6	78.6						

## 2. MODELING TECHNIQUES

### 2.1 The Grey Relational Analysis

This is comparatively a novel technique in software estimations and can be effectively used for analyzing the relationships between two series. It is a technique of the Grey Systems Theory (GST), which was introduced by Deng [1,2,3]. The term **Grey** lies between *Black* (meaning “no information”) and *White* (meaning “full information”) and it indicates that the information is partially available. It is suitable for unascertained problems with poor information. GRA has the ability to learn from a small number of cases, which is effective in the context of data-starvation [17] The magnetism of GRA is its flexibility to model a complex nonlinear relationship [18,20]. The basic concepts of GRA are as given below:

*Factor space:*

$\{p(X); Q\}$  is a factor space where  $p(X)$  is a theme described by the factor set  $X$ , and  $Q$  as the influence relation. The factor space  $\{p(X); Q\}$  has the following properties[17]:

1. Existence of key factors.
2. The number of factors is limited and countable.
3. Factor independence.
4. Factor expansibility.

Suppose  $x_i = \{x_i(1), x_i(2), \dots, x_i(m)\}$ , where,  $i = 0, 1, 2, \dots, n \in N; m \in N$ , is a data series.

*Comparable series:*

The series is *comparable* if, and only if, they are *dimensionless, scaled, and polarized*.

*Grey relational space:*

The factor space is a grey relational space if all the series in a factor space  $\{p(X); Q\}$  are comparable. It is denoted as  $\{p(X); \Gamma\}$ . In a grey relational space  $\{p(X); \Gamma\}$ ,  $X$  is a collection of data series  $x_i(i = 0, 1, \dots, n)$ , in which  $x_i = \{x_i(1), x_i(2), \dots, x_i(k)\}$ , is the series; and  $k = 1, 2, \dots, m$ , are the factors.  $\Gamma$ , which is the Grey Relational Map set. It is based on geometrical mathematics and has the following four properties [17]:

Normality  $0 \leq \Gamma(x_i(k), x_j(k)) \leq 1, \forall i, \forall j, \forall k,$   
 $\Gamma(x_i, x_j) = 1 \Leftrightarrow x_i \equiv x_j,$   
 $\Gamma(x_i, x_j) = 0 \Leftrightarrow x_i \cap x_j \in \phi$

Symmetry  $\forall x_i, \forall x_j \in X,$   
 $\Gamma(x_i, x_j) = \Gamma(x_j, x_i) \Leftrightarrow X = \{x_i, x_j\}.$

Entirety  $\forall x_i, \forall x_j \in X = \{x|\sigma = 0, 1, \dots, n\}, n \geq 2,$   
 $\Gamma(x_i, x_j) \text{ often } \neq \Gamma(x_j, x_i).$

Proximity  $\Gamma(x_i(k), x_j(k)) \text{ increases as } \Delta(k) = |x_i(k) - x_j(k)| \text{ decrease for } \forall k \in \{1, 2, \dots, m\}.$

2.1.1 Grey Relational Grade by Deng's Method

GRA is used to measure all the influences of various features and the relationship among data series that is a collection of measurements [1-3]. The steps involved are as listed below.

Data Processing

The first step is the standardization of the various attributes. Every attribute has the same amount of influence, as the data is made dimensionless by using various techniques like upper bound effectiveness, lower bound effectiveness, or moderate effectiveness. The formulas for data processing techniques are as follows:

- Upper-bound effectiveness (i.e., the larger the better)

$$x_i^*(k) = \frac{x_i(k) - \min_i x_i(k)}{\max_i x_i(k) - \min_i x_i(k)} \tag{1}$$

where  $i=1,2,\dots,m$  and  $k=1,2,\dots,n$ .

- Lower-bound effectiveness (i.e., the smaller the better)

$$x_i^*(k) = \frac{\max_i x_i(k) - x_i(k)}{\max_i x_i(k) - \min_i x_i(k)} \tag{2}$$

where  $i=1,2,\dots,m$  and  $k=1,2,\dots,n$ .

- Moderate effectiveness (i.e., the nominal is the best)

$$x_i^*(k) = 1 - \frac{|x_i(k) - x_{ab}(k)|}{\max\{\max_i x_i(k) - x_{ab}(k), x_{ab}(k) - \min_i x_i(k)\}} \tag{3}$$

where  $i=1,2,\dots,m$  and  $k=1,2,\dots,n$ .

$x_i(k)$  represents the value of the  $k_{th}$  attribute in the  $i_{th}$  series;  $(k)$  represents the modified value of the  $k_{th}$  attribute in the  $i_{th}$  series;  $\max_i x_i$  represents the maximum of the  $k_{th}$  attribute in all series;  $\min_i x_i$  represents the minimum of the  $k_{th}$  attribute in all series, and  $x_{ab}(k)$  is the objective value of the  $k_{th}$  attribute.

Difference Series

The GRA uses the grey relational coefficient to describe the trend relationship between an objective series and a reference series at a given point in a system.

$$\gamma(x_0(k), x_i(k)) = \frac{\Delta_{\min} + \zeta\Delta_{\max}}{\Delta_{o,i}(k) + \zeta\Delta_{\max}} \tag{4}$$

where  $\Delta_{0,i}(k) = |x_0(k) - x_i(k)|$  is the difference of the absolute value between  $x_0(k)$  and  $x_i(k)$ ;  
 $\Delta_{min} = \min_j \min_k |x_0(k) - x_j(k)|$  is the smallest value of  $\Delta_{0,j} \forall j \in \{1, 2, \dots, n\}$ ;  
 $\Delta_{max} = \max_j \max_k |x_0(k) - x_j(k)|$  is the largest value of  $\Delta_{0,j} \forall j \in \{1, 2, \dots, n\}$ ;  
 and  $\zeta$  is the distinguishing coefficient,  $\zeta \in (0, 1]$ .

The  $\zeta$  value will change the magnitude of  $\gamma(x_0(k), x_i(k))$ . In this study the value of  $\zeta$  has been taken as 0.5 Deng [23].

*Grey Relational Grade*

The GRG is used to find the overall similarity degree between reference tuple  $x_0$  and comparative tuple  $x_i$ . When the value of the GRG approaches 1, the two tuples are “more closely similar.” When the GRG approaches a value of 0, the two tuples are “more dissimilar.” The GRG  $\Gamma(x_0, x_i)$  between an objective series  $x_i$  and the reference series  $x_0$  was defined by Deng as follows:

$$\Gamma(x_0, x_i) = \frac{1}{n} \sum_{k=1}^n \gamma(x_0(k), x_i(k)) \tag{5}$$

**2.2 Regression**

Regression analysis is a statistical technique for modelling and analysing variables. It is used to study the relationship that exists between a dependent variable and one or more independent variables.

*2.2.1 Ordinary Least Square Regression*

It is the most popular and widely applied technique to build software cost estimation models. According to the principle of least squares the “best fitting” line is the line which minimizes the deviations of the observed data away from the line. The regression parameters for the least square line, are the estimates of the unknown regression parameters in the model. This is also referred to as multiple linear regression and is given by:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \dots + \beta_k x_{i,k} + \varepsilon_i \tag{6}$$

where,  $Y_i$  is a dependent variable whereas  $x_1, x_2, \dots, x_k$  are  $k$  independent variables.  $\beta_0$  is the  $y$  intercept,  $\beta_1, \beta_2$  are the slope of  $y$ ,  $\varepsilon_i$  is the error term. The corresponding prediction equation is given as:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i,1} + \dots + \hat{\beta}_k x_{i,k} \tag{7}$$

In this equation  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k$  are the least square coefficients and  $\hat{y}_i$  is the estimated response for the  $i^{th}$  term.

Thus, the response estimated from the regression line minimizes the sum of squared distances between the regression line and the observed response.



### 2.2.2 Robust Regression

Robust Regression is a type of regression technique, which prevails over the limitations of the ordinary least square. Ordinary least square estimates are extremely non-robust to outliers. Outliers are those observations in the dataset that do not follow the prototype of the other observations. These can inefficiently influence the whole process of fitting. In this study, we evaluated the performance of some well-known robust M-estimators. They are as follows: the “Andrew-estimator” [40], the “Bisquare-estimator,” the “Cauchy-estimator,” the “Fair-estimator,” the “Huber-estimator” [38, 39, 43], the “Logistic-estimator,” the “Talwar-estimator,” and the “Welsch-estimator,” etc. They have been used with their different weight functions [41] for predicting the software effort of the projects.

M-estimators are also called maximum-likelihood estimators, as they try to minimize the weighted sum of residuals. They work on the principle of the Iteratively Reweighted Least Square (IRLS). The least square method tries to minimize  $\sum e_i^2$ , which is unstable in case there are outliers present in the data, whereas the M-estimators try to minimize the effect of these outliers by substituting the squared residuals  $e_i$  by the function given below.

$$\min \sum_{i=1}^n \rho(e_i) \quad (8)$$

The steps involved in IRLS [42] are:

- Step 1:* In the first iteration, each observation is allocated an equal weight and the coefficients of the model are estimated using OLS.
- Step 2:* In the second step, after the OLS, residuals are used to find weights. The observation with a larger residual is assigned a lower weight.
- Step 3:* In the third iteration, the new model parameters and the residuals are recomputed using Weighted Least Squares (WLS).
- Step 4:* In Step 4, new weights, as per Step 2, are found and the procedure continues until the values of the parameter estimates converge within a specified tolerance.

### 2.2.3 Stepwise Regression

This is a method for adding and removing terms based on their statistical importance. The forward approach starts with no variables in the model. It tries out the variables one by one and includes them if they are “statistically important.” The selection has been used for estimating the effort of the reference project for various similar projects. At each step, a predictor is entered based on partial  $F$ -tests or  $t$  test. The procedure continues until more variables can be justifiably entered. The first variable that is put in the stepwise model is the variable having the smallest  $t$ -test  $P$ -value (below  $\alpha_E = 0.05$ ). The level of significance ( $\alpha$ ) is taken to be 5%.

## 3. PROPOSED ALGORITHM, FOR SOFTWARE ESTIMATION

### 3.1 Modeling the Grey Relational Effort Analysis Technique with Regression Methods (GREAT\_RM)

In this methodology, the focus is project selection based on GRA and effort prediction by re-

gression. In the GRA based studies so far, effort is estimated by generating similar projects to the target project and then estimating effort from those most similar projects. *GREAT\_RM* uses GRA for generating similar projects to the target project, but effort is calculated by applying regression on those most similar projects. In this study, ten regression techniques have been applied, including OLS, eight robust regression methods, and SWR. The individuality of the work is that the value of  $k$  varies with each reference project. The basic steps of the methodology are:

*Step1: Select the continuous attributes from the dataset.*

*Step2: Data series construction:* The data set consists of series  $x_0 = \{x_1(1), x_1(2), \dots, x_1(m)\}$ ,  $x_1 = \{x_2(1), x_2(2), \dots, x_2(m)\}$ ,  $x_2 = \{x_3(1), x_3(2), \dots, x_3(m)\}$  and  $x_n = \{x_n(1), x_n(2), \dots, x_n(m)\}$ ,  $x_0$  is the reference series whose effort is to be estimated based on the objective series  $x_1, x_2, \dots, x_n$ .

*Step3: Data Preparation:* The numerical features are normalized in a specified range so that each feature has the same weight on effort and hence it eases out their comparisons and processing.

*Step4: Ranking the  $k$  closest projects:* This aims at retrieving software projects from the data that exhibits large similarities with projects that are under investigation. The distance  $\overline{\Delta}_{o,i}$  between two tuples at the  $k^{th}$  feature, is calculated by the formula as shown in Eq. (4).

$$\gamma(x_0(k), x_i(k)) = \frac{\Delta_{\min} + \zeta \Delta_{\max}}{\Delta_{o,i}(k) + \zeta \Delta_{\max}} \tag{9}$$

For all grey relational grades between the reference project  $o$  and the  $i^{th}$  comparative project, the  $\Gamma(x_o, x_i)$  values are calculated for each  $i$  according to Eq. (5). The range of  $\Gamma$  is from 0 to 1 in each case [47]. For more similarity between projects the value of  $\Gamma$  approaches one and for two projects that are completely dissimilar, it approaches zero. The projects that have the higher value on GRG get the greatest opportunity to contribute in the final estimate.

*Step5. Effort Prediction by GRA:* The effort for GRA is the simple aggregation of the  $k$  most influential projects [17].

$$\hat{\epsilon} = \sum_{i=1}^k w_i * \epsilon_i \tag{10}$$

where,  $\epsilon_i$  is the effort of the  $i^{th}$  most influential project and  $w_i$  is the weight given by:

$$w_i = \frac{\tau(x_0, x_i)}{\sum_{j=1}^k \tau(x_0, x_j)} \tag{11}$$

*Step 6. Effort Prediction by Regression:* In this step, effort estimate for a given project is cal-

culated by applying various regression techniques on only the  $k$  most similar projects obtained from Step 4.

### 3.2 Evaluation criteria

For the purpose of validating and evaluating the new methodology, the basic necessity is to measure how accurate the estimations are. There are various approaches used by researchers to measure the accuracy of effort prediction methods, such as the Mean Magnitude of Relative Error (MMRE), the Median Magnitude of Relative Error (MdMRE), the Magnitude of Relative Error Relative to the Estimate (MMER), the adjusted R-squared or coefficient of determination, and  $Pred(n)$ . In order to measure the accuracy of the software estimation, we have used the four most popularly used evaluation criteria in software engineering, (i.e., MMRE, MdMRE, MMER and  $Pred(n)$ .)

#### 3.2.1 Mean Magnitude of Relative Error (MMRE)

Relative error is the absolute error in the observations divided by its actual value. The magnitude of relative error is the percentage of actual effort for the project and is given as:

$$MRE = \frac{|actual - estimated|}{actual} \quad (12)$$

The MRE for each observation is aggregated over the total number of projects,  $N$ , in order to generate the mean MRE (MMRE). MMRE is calculated as:

$$MMRE = \frac{1}{N} \sum_{x=1}^N MRE_x \quad (13)$$

MMRE favors models that underestimate, and it is extremely sensitive to small actuals. As MMRE is sensitive to individual predictions with excessively large MREs, we also use the median of MRE's for the  $n$  projects. (MdMRE), which is less sensitive to extreme values, is used as another measure for estimation accuracy.

#### 3.2.2 Median MRE (MdMRE)

Median MRE is less sensitive to extreme values as compared to MMRE, so in case of large datasets we prefer using MdMRE as the estimation accuracy criteria and it is given by:

$$MdMRE = median (MRE_x) \quad (14)$$

A higher score for both MMRE and MdMRE means worse prediction accuracy.

#### 3.2.3 Magnitude of Relative Error relative to the Estimate (MMER):

Another measure akin to MRE is MER. It is more preferable to MRE since it measures the error relative to the estimate. MER is given by:

$$MER = \frac{|actual_i - estimated_i|}{estimated_i} \tag{15}$$

We use the notation MMER to denote the mean MER.

$$MMER = \frac{1}{N} \sum_{x=1}^N MER_x \tag{16}$$

### 3.2.4 Pred (n)

It is used as the opposite measure to count the percentage of estimates that fall within less than *n* percent of the actual values. The common used value for Pred(*n*) is 25% . A low score on MMRE, MdmRE, MMER and a high score on Pred (*n*) entails better accuracy.

## 3.3 Data Sources

The data used in the present study comes from the PROMISE repository [44]. Though these datasets are old, they are still extensively being used to assess the comparative accuracy of new techniques. The descriptive statistics of these sets are given below. Five different datasets have been used in the study with the belief that if the model validation is successful with these models then it can also be validated on any other dataset. The descriptive statistics of all the five dataset are given below in Table 2.

Table 2. Descriptive Statistics of the Datasets

	<i>Dataset</i>	<i>Cases</i>	<i>Features</i>	<i>Effort Mean</i>	<i>Effort Standard Deviation</i>
1.	Finnish	38	8	7,678.29	7,135.28 (hours)
2.	Desharnais	77	9	4,834	4,188 (hours)
3.	COCOMO-81	63	17	683.52	1,821.51 (hours)
4.	Albrecht	24	8	21,875	28,417 (hours)
5.	Kemerer	15	5	219.25	263 (man hours)

## 4. EXPERIMENTAL RESULTS

The results obtained using the proposed methods are summarized in Table 3. The table provides the comparison of results obtained using only GRA, GRA and OLS, GRA and RR, and finally GRA and SWR for all of the five datasets. Only the best result obtained using eight robust regression techniques have been produced and put to statistical analysis. This empirical study is carried out using Leave One Out Cross Validation (LOOCV). For each iteration, one project is held out once and the training is performed on (*n*-1) projects. The accuracy is then measured by aggregating the accuracy of all the sets.

Boxplot of absolute residuals and Wilcoxon signed rank test of residuals are applied in order to study the distribution of residuals and statistical significance of *GREAT\_RM* technique.

Table 3. Prediction Accuracy Results of GREAT\_RM

	GRA	GRA with OLS	GRA with Robust Regression	GRA with Stepwise
<i>Finnish dataset</i>				
MMRE	11.37	11.88	10.3	58.61
Median (MRE)	2.67	2.19	1.93	27.93
MMER	12	12.88	9.81	40.99
Pred (25)	76.32	89.47	89.47	47.37
<i>Albrecht dataset</i>				
MMRE	46.35	29.83	24.16	32.64
Median (MRE)	4.89	7.47	10	12.39
MMER	17.15	21.72	22.57	24.87
Pred (25)	70.83	70.83	70.83	70.83
<i>COCOMO -81 dataset</i>				
MMRE	30	32.35	25.59	21.04
Median (MRE)	7.07	5.32	4.82	9.42
MMER	26.86	15.71	18.9	48.71
Pred (25)	68.25	76.19	74.6	76.19
<i>Desharnais dataset</i>				
MMRE	34.9	18.19	25.44	16.78
Median (MRE)	5.07	1.51	8.36	7.84
MMER	22.12	8.74	18.97	31.63
Pred (25)	68.83	90.9	79.22	74.02
<i>Kemerer dataset</i>				
MMRE	46.67	35.18	29.63	38.65
Median(MRE)	11.46	21.92	16.56	31.35
MMER	35.58	41.04	27.93	78.8
Pred (25)	60	53.33	60	40

Boxplots represent the results obtained by GRA, GRA+OLS, GRA+RR, and GRA+SWR.

1. The Boxplots in Fig. 1,2,3,4, and 5 show that medians are very close to zero for all the datasets, thus indicating that the spread is tighter towards the minimum value. The box overlays the lower tail for Albrecht, COCOMO-81, Desharnais, and Kemerer datasets, thus presenting an accurate prediction.
2. The range of absolute residuals for the Finnish dataset is large for GRA and GRA+SWR, but because the medians are tilted towards the minimum value, it indicates that the predictions are good.

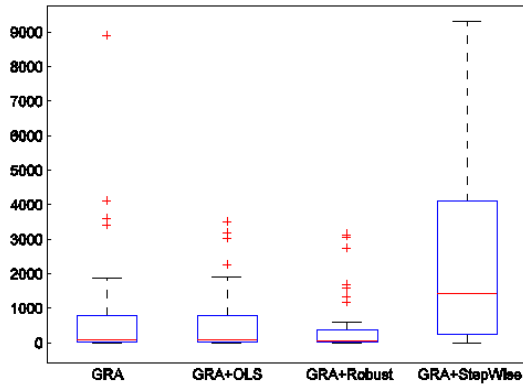


Fig. 1. Boxplot of Absolute Residuals for the Finnish dataset

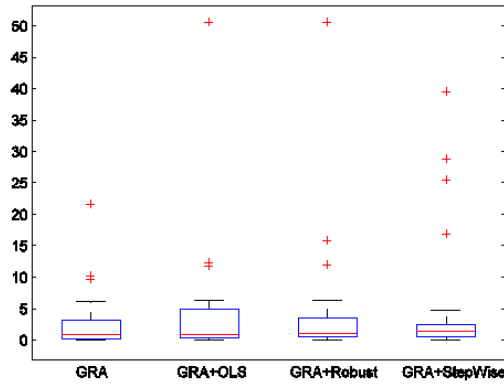


Fig. 2. Boxplot of Absolute Residuals for the Albrecht dataset

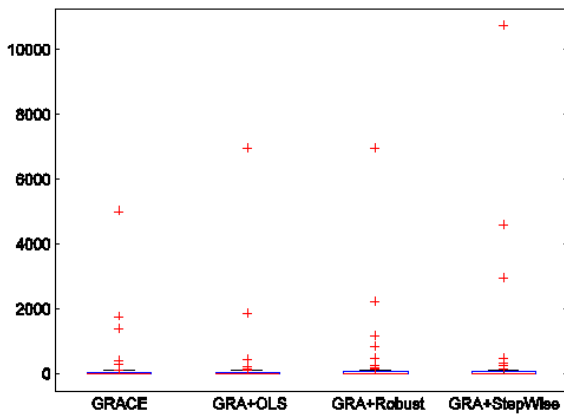


Fig. 3. Boxplot of Absolute Residuals for the COCOMO-81 dataset

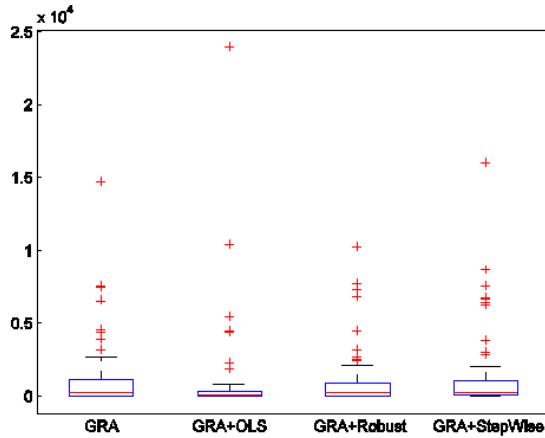


Fig. 4. Boxplot of Absolute Residuals for the Desharnais dataset

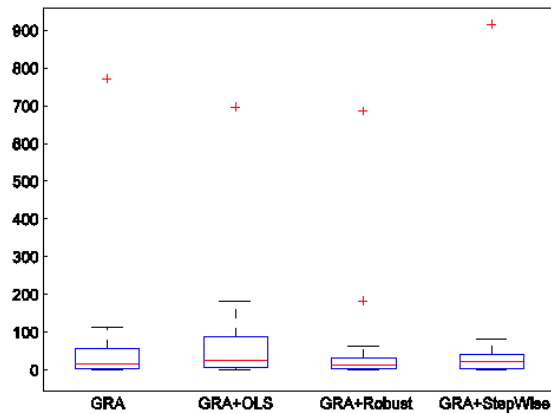


Fig. 5. Boxplot of Absolute Residuals for the Kemerer Dataset

#### 4.1 Comparison of GREAT\_RM to Multiple Linear Regression, Ridge Regression, Neural Networks, Support Vector Machine, and Multiple Adaptive Regression Splines

This section presents the results obtained when we compared the *GREAT\_RM* model to Multiple Linear Regression (MLR), Ridge Regression (RR), Artificial Neural Networks (ANN), the Support Vector Machine (SVM), and Multiple Adaptive Regression Splines (MARS). We should note here that all the techniques were applied to the same numerical features of the same five datasets. However, in this section we investigate whether these techniques produce equivalent or better results than *GREAT\_RM*. We used boxplots of absolute residuals to statistically measure the distribution of residuals. Based on absolute residuals we tested the statistical significance of all the results. All statistical significant tests were obtained using SPSS 19 for Windows.

(1) *Multiple Linear Regression (MLR)*: In this work, the MLR uses data collected from historical projects to examine the relationships between independent attributes and dependent attributes and then developed a formal model based on that. This is already explained in Section 2.2.1

(2) *Artificial Neural Network (ANN)*: ANNs are very sophisticated modeling and prediction techniques that are capable of modeling extremely complex functions and data relationships. They have the ability to learn by examples, which enables the user to model data and establish accurate rules governing the underlying relationship between various data attributes. They take the sample data, and then invoke *training algorithms*, which can automatically learn the structure of the data. One of the abilities of the neural network is to accurately predict data that is not part of the training dataset, and the process is known as *generalization*. In this work, multilayer perceptrons have been applied. Multilayer perceptrons are supervised feed-forward networks trained with a back propagation algorithm. With training input data and preferred output data, the multilayer perceptrons are trained on how to convert input data into a particular output. Three parameters have a major impact on the accuracy of the network, which should be defined before building the network. They are: number of hidden layers, the number of neurons in each hidden layer, and the type of activation functions [23].

(3) *Ridge Regression (RR)*: RR is an alternative regression technique that tries to address the potential problems with OLS that arise due to highly correlated attributes. In regression, the objective is to “explain” the variation in one or more “response variables,” by associating this variation with proportional variation in one or more “explanatory variables. However, the problem arises when the explanatory variables vary in similar ways, which reduces their collective power of explanation. This phenomenon is known as *near collinearity*. As the different variables are correlated the covariance matrix  $X'X$  will be nearly singular and as a result the estimates will be unstable. A small variation in error will have a large impact on  $\hat{\beta}$ . Ridge regression reduces the sensitivity by adding a number  $\delta$  to the elements on the diagonal of the matrix that is to be inverted.  $\delta$  is called the ridge parameter and it yields the following estimator of  $\beta$ .

$$\hat{\beta}_{\delta} = (X'X + \delta I_n)^{-1}(X'e) \tag{17}$$

where,  $I_n$  represents the identity matrix of rank  $n$ .

(4) *Support Vector Machine (SVM)*: The regression SVM estimates the functional dependence of the dependent variable  $y$  on a set of independent variables  $x$ . It assumes, like other regression problems, that the relationship between the independent and dependent variables is given by a deterministic function of  $f$  plus the addition of some additive noise.

$$y = f(x) + \text{noise} \tag{18}$$

SVM finds a functional form for  $f$  that correctly predicts new cases that are presented to SVM. This is achieved by training the SVM model on a sample set (i.e., training set, a process that involves the sequential optimization of an error function). SVM constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space, which can be used for regression. A



good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class. In general, the larger the margin, the lower the generalization error of the classifier. To construct an optimal hyper plane, SVM employs an iterative training algorithm, which minimizes an error function.

(5) *MAR Splines*: MAR Splines focuses on the development and deployment of accurate and easy-to-understand regression models. The MAR Splines model is designed to predict continuous numeric and high quality probability models. The MAR Splines model is a regression model that automatically generates non-linearities and interactions between variables and is thus a promising technique to be used for software effort estimation. MAR Splines has shown evidences of very high-performance results in forecasting the electricity requirement for power-generating companies, relating customer satisfaction scores to the engineering specifications of products, and presence/absence modeling in Geographical Information Systems (GIS). MAR Splines fits the data in the following equation:

$$e_i = b_0 + \sum_{k=1}^K b_k \prod_{i=1}^L h_i(x_i(j)) \tag{19}$$

In this  $b_0$  and  $b_k$  are the intercept and slope. Parameters  $h_i(x_i(j))$  are the hinge functions. They take the form  $\max(0, x_i(j)-b)$  where,  $b$  is the knot. MAR Splines behaves as a multiple piece wise linear regression by adding multiple hinge functions.

#### 4.2 Comparison of the Finnish Dataset

Table 4 summarizes the accuracy of the respective methods when applied to the Finnish data. From the results we can observe that *GREAT\_RM* produced better accuracy than all other models.

The results also revealed that *GREAT\_RM*, ANN, and MAR Splines are similar in terms of MMRE, MdMRE, and Pred (25) accuracy. The Boxplot of absolute residuals for the various techniques shows that:

- The box of *GREAT\_RM* overlays the lower tail, which implies that the absolute residuals are skewed towards the minimum value and that it also presents a more accurate estimation than all other models.
- The median of *GREAT\_RM* is smaller than the median of other models, which revealed that at least half of the predictions are more accurate than other models.

Table 4. Results of the Finnish Dataset

	OLS	RR	ANN	SVM	MAR Splines	GREAT_RM
Finnish Dataset						
MMRE	0.75	0.71	0.06	0.42	0.08	0.103
MdMRE	0.36	0.32	0.02	0.16	0.07	0.193
MMER	0.05	1.3	0.05	0.13	0.09	0.981
Pred (25)	36.84	36.84	92.11	63.15	97.37	89.47

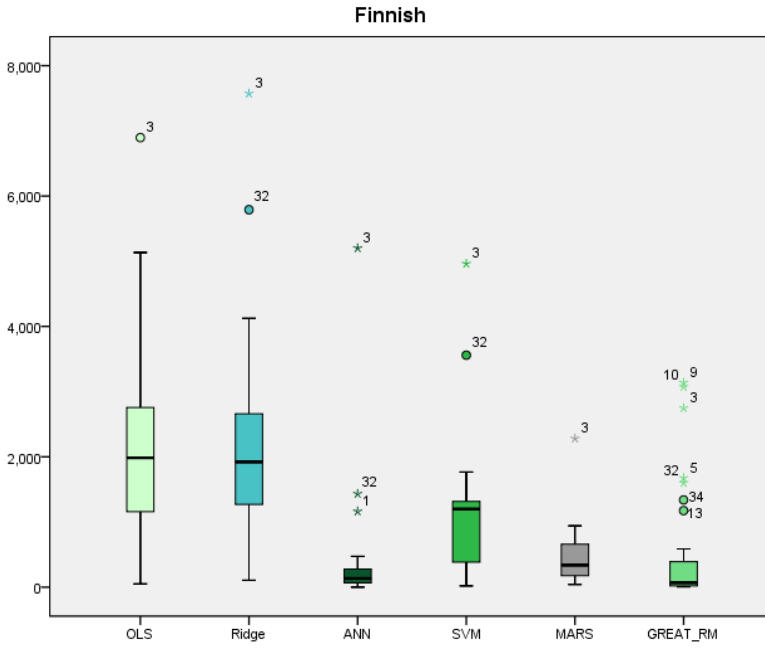


Fig. 6. Boxplot of Absolute Residuals for Finnish data set

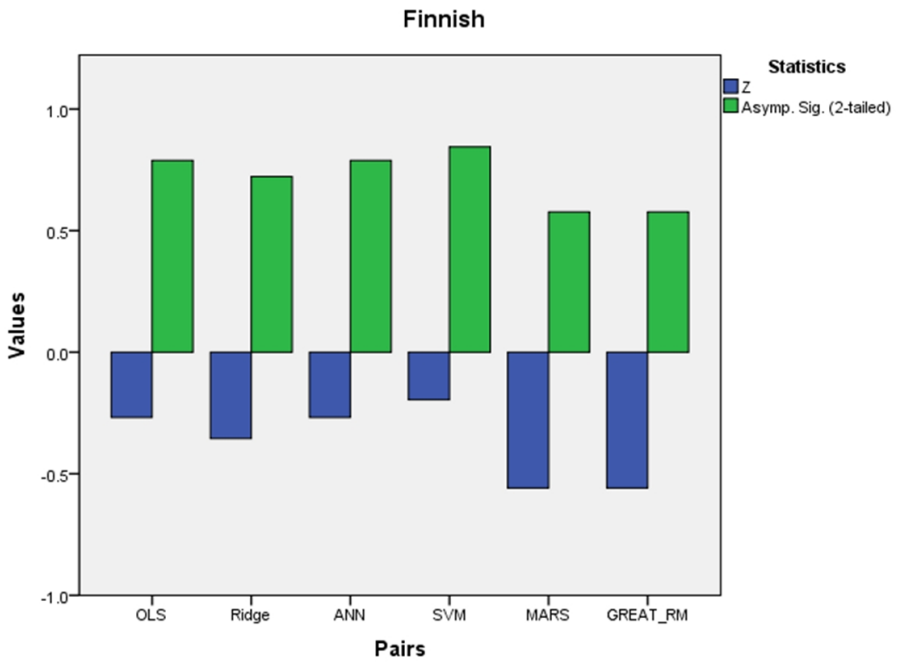


Fig. 7. Bar Graph(WILCOXON Signed Rank Test)

Table 5. Wilcoxon signed rank test (Test Statistics<sup>c</sup>)

Finnish Dataset	OLS	Ridge	ANN	SVM	MARS	GREAT_RM
Z	-.268 <sup>a</sup>	-.355 <sup>a</sup>	-.268 <sup>b</sup>	-.196 <sup>b</sup>	-.558 <sup>a</sup>	-.558 <sup>a</sup>
Asymp. Sig. (2-tailed)	.788	.722	.788	.845	.577	.577

a. Based on negative ranks. b. Based on positive ranks. c. Wilcoxon signed rank test

*WILCOXON Signed Rank Test:*

Step 1: Hypotheses

$$H_0: \mu (\text{Before}) = \mu (\text{After}) ; H_a: \mu (\text{Before}) > \mu (\text{After})$$

(Residual Median) = Hypothetical Median (test value).

Step 2: Significance Level:  $\alpha = 0.05$ .

Step 3: Rejection Region: Reject the null hypothesis if p-value  $\leq 0.05$ .

Step 4: Test Statistic: Wilcoxon signed rank test.

Step 5: Decision.

The p-value in all the cases is greater than 0.05, as shown in Table 5. Thus, we accept the null hypothesis. Consequently, we conclude that the residuals obtained by using all approaches were not significantly different from the test value of zero. As a result, the proposed methods can be used for software effort estimation.

### 4.3 Comparison of the Albrecht Dataset

On analyzing the Albrecht results, Table 6 summarizes the accuracy of the respective methods. From the results we can observe that GREAT\_RM produced better accuracy than OLS, ridge, ANN, SVM, and MAR Splines.

The results also revealed that GREAT\_RM produced better results of not only MMRE, but also MdMRE, MMR, and Pred (25). Thus, GREAT\_RM tends to be more accurate than all other models. Amongst the other models, ANN produced better results in the Albrecht dataset. Unsurprisingly, predictions based on the GREAT\_RM model presented statistically significant accurate estimations, which was also established by the Boxplot of absolute residuals as shown in Fig. 8.

The Boxplot of absolute residuals for the various techniques shows that:

- The box of GREAT\_RM overlays the lower tail, which shows that the absolute residuals are skewed towards the minimum value and also presents an accurate estimation.
- The range of absolute residuals of GREAT\_RM and ANN are smaller than the absolute residuals of other models, which implies that the variance is less.

Table 6. Results of the Albrecht Dataset

	OLS	RR	ANN	SVM	MAR Splines	GREAT_RM
Albrecht Dataset						
MMRE	0.9	0.91	0.78	0.51	1.23	0.24
MdMRE	0.43	0.52	0.17	0.33	0.6	0.10
MMR	0.17	0.27	0.28	0.35	0.52	0.23
Pred (25)	37.5	37.5	58.33	45.83	29.17	70.83

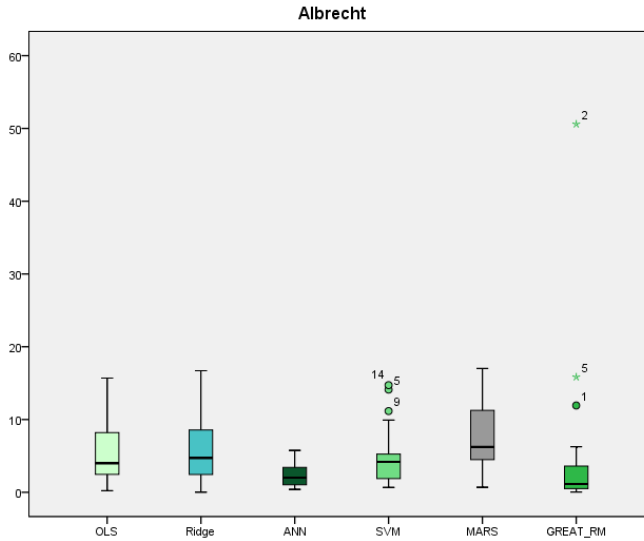


Fig. 8. Boxplot of Absolute Residuals for Albrecht data set

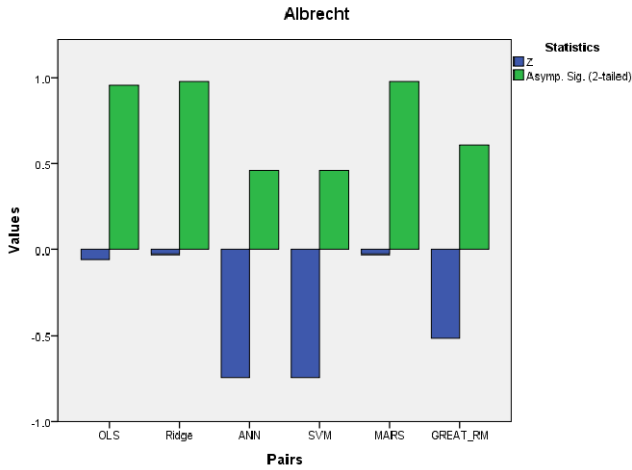


Fig. 9. Bar Graph (WILCOXON Signed Rank Test)

- The median of *GREAT\_RM* is smaller than the median of other models, which revealed that at least half of the predictions of FGRA are more accurate than other models.

*WILCOXON Signed Rank Test:*

The p-value in all the cases is greater than 0.05, as shown in Table 7. Thus, we accept the null hypothesis. Consequently, we conclude that the residuals obtained by using all approaches were not significantly different from the test value of zero. As a result, the proposed methods can be used for software effort estimation.

Table 7. Wilcoxon signed rank test (Test Statistics<sup>c</sup>)

Albrecht Dataset	OLS	Ridge	ANN	SVM	MARS	GREAT_RM
Z	-.057 <sup>a</sup>	-.029 <sup>a</sup>	-.743 <sup>a</sup>	-.743 <sup>b</sup>	-.029 <sup>b</sup>	-.514 <sup>a</sup>
Asymp. Sig. (2-tailed)	.954	.977	.458	.458	.977	.607

a. Based on positive ranks.      b. Based on negative ranks.      c. Wilcoxon signed rank test

#### 4.4 Comparison of the COCOMO-81 Dataset

Table 8 summarizes the accuracy of the various models in the COCOMO\_81 dataset. From the results we can observe that *GREAT\_RM* performed significantly better than all of the other models. The results also revealed that *GREAT\_RM* produced credible results in terms of not only MMRE but also MdMRE, MMR, and Pred (25).

The Boxplot of absolute residuals for the various techniques shows that:

Predictions based on the *GREAT\_RM* model presented statistically significant accurate estimations, measured using absolute residuals, and was confirmed by the results of the Boxplot of absolute residuals, as shown in Fig. 10.

- The box of *GREAT\_RM* has been minimized as can be seen in Fig. 10, which shows that

Table 8. Results of the COCOMO\_81 Dataset

	OLS	RR	ANN	SVM	MAR Splines	GREAT_RM
COCOMO-81 Dataset						
MMRE	14.77	12.73	2.67	11.25	16.79	0.25
MdMRE	3.72	3.56	0.88	2.39	2.73	0.482
MMR	0.33	0.52	1.77	0.18	15.87	0.189
Pred (25)	11.11	12.69	25.39	12.69	0.02	74.6

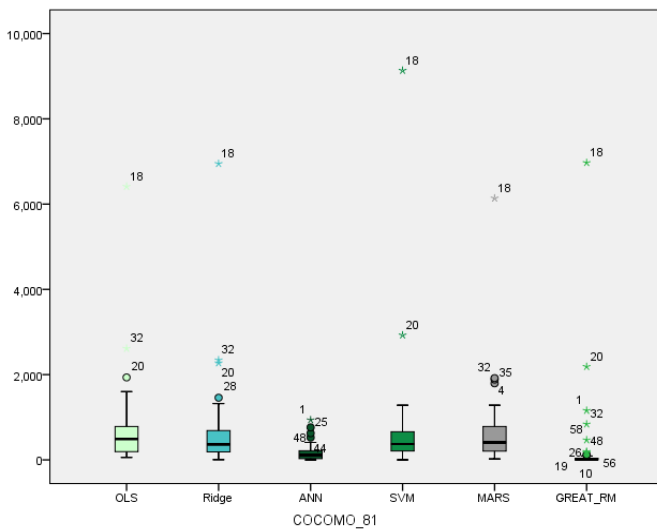


Fig. 10. Boxplot of Absolute Residuals for the COCOMO\_81 Dataset

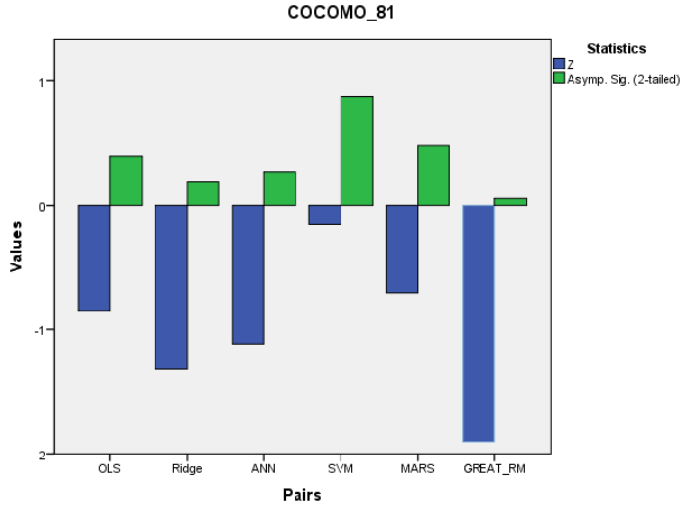


Fig. 11. Bar Graph (WILCOXON Signed Rank Test)

Table 9. Wilcoxon signed rank test (Test Statistics)

COCOMO_81 Dataset	OLS	Ridge	ANN	SVM	MARS	GREAT_RM
Z	-.849 <sup>a</sup>	-1.321 <sup>a</sup>	-1.116 <sup>a</sup>	-.157 <sup>a</sup>	-.705 <sup>a</sup>	-1.903 <sup>a</sup>
Asymp. Sig. (2-tailed)	.396	.186	.264	.875	.481	.057

a. Based on positive ranks. b. Wilcoxon signed rank test

the absolute residuals are skewed towards the minimum value and thus presents a more accurate estimation than the other five models.

- The range of absolute residuals of *GREAT\_RM* is smaller than the absolute residuals of others.
- The median of *GREAT\_RM* is approaching zero, which revealed that at least half of the predictions of FGRA are more accurate than other models.
- On the other hand, the range of absolute residual values for OLS, Ridge, SVM, and MARS produced the worst individual estimation.

*WILCOXON Signed Rank Test:*

The p-value in all the cases is greater than 0.05, as shown in Table 9. Thus, we accept the null hypothesis. Consequently, we conclude that the residuals obtained by using all approaches are not significantly different from the test value of zero. As a result, all the proposed methods can be used for software effort estimation.

4.5 Comparison of the Desharnais Dataset

Table 10 summarizes the accuracy of the various models in the Desharnais dataset. From the results we can observe that *GREAT\_RM* performed significantly better than all of the other models. The results also revealed that *GREAT\_RM* produced credible results in terms of not

Table 10. Wilcoxon Signed Rank Test

	OLS	RR	ANN	SVM	MAR Splines	GREAT_RM
Desharnais Dataset						
MMRE	0.5	0.47	0.47	0.48	0.51	0.16
MdMRE	0.31	0.3	0.31	0.31	0.32	0.078
MMER	0.40	0.41	0.35	0.16	0.40	0.31
Pred (25)	35.06	41.56	31.63	40.26	35.06	74.19

only MMRE but also MdMRE and Pred (25).

The Boxplot of absolute residuals for the various techniques shows that:

- Predictions based on the *GREAT\_RM* model presented statistically significant accurate estimations, measured using absolute residuals, and was confirmed by the results of the boxplot of absolute residuals, as shown in Fig 12.
- The medians for the *GREAT\_RM* techniques applied to the Desharnais dataset are very close to zero, as is clear from the values on the Y-axis, indicating that the estimates were closer to the minimum value.
- The range of absolute residuals of *GREAT\_RM* is smaller.
- On the other hand, the range of absolute residual values for OLS, Ridge, ANN, SVM, and MARS produced the worst individual estimations.
- Extreme values presented in other models affected the estimation process.

*WILCOXON Signed Rank Test:*

The p-value in all the cases is greater than 0.05, as shown in Table 11. Thus, the proposed methods are statistically significant. As a result, all of the proposed methods can be used for software effort estimation.

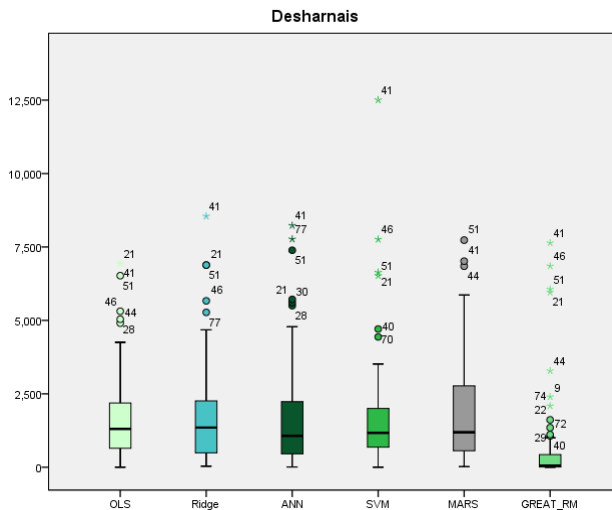


Fig. 12. Boxplot of Absolute Residuals for the Desharnais Dataset

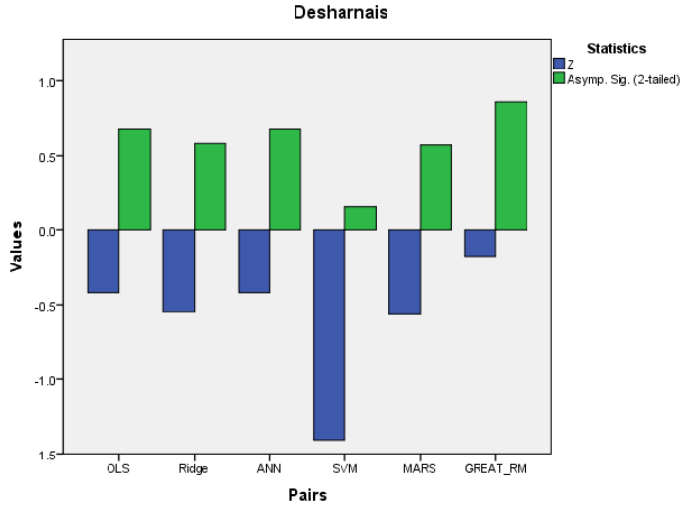


Fig. 13. Bar graph(WILCOXON Signed Rank Test)

Table 11. Wilcoxon Signed Rank Test (Test Statistics<sup>c</sup>)

Desharnais Dataset	OLS	Ridge	ANN	SVM	MARS	GREAT_RM
Z	-.419 <sup>a</sup>	-.551 <sup>a</sup>	-.419 <sup>a</sup>	-1.409 <sup>a</sup>	-.566 <sup>a</sup>	-.175 <sup>b</sup>
Asymp. Sig. (2-tailed)	.675	.582	.675	.159	.571	.861

a. Based on positive ranks. b. Based on negative ranks. c. Wilcoxon Signed Rank Test

#### 4.6 Comparison of the Kemerer Dataset

Table 12 summarizes the accuracy of the various models in the Kemerer dataset. From the results we can observe that *GREAT\_RM* and ANN performed significantly better than all other models. The results also revealed that both the techniques produced credible results in terms of not only MMRE but also MdMRE, MMER, and Pred (25).

The Boxplot of absolute residuals for the various techniques shows that:

- Predictions based on the *GREAT\_RM* model presented statistically significant accurate estimations, measured using absolute residuals, and was confirmed by the results of boxplot of absolute residuals, as shown in Fig 14.

Table 12. Wilcoxon signed rank test

	OLS	RR	ANN	SVM	MAR Splines	GREAT_RM
Kemerer Dataset						
MMRE	0.74	0.68	0.31	0.003	1.05	0.296
MdMRE	0.54	0.37	0.12	0.0012	0.65	0.16
MMER	0.028	0.04	0.37	8.47	0.25	0.28
Pred (25)	26.67	20	66.67	100	3.33	60



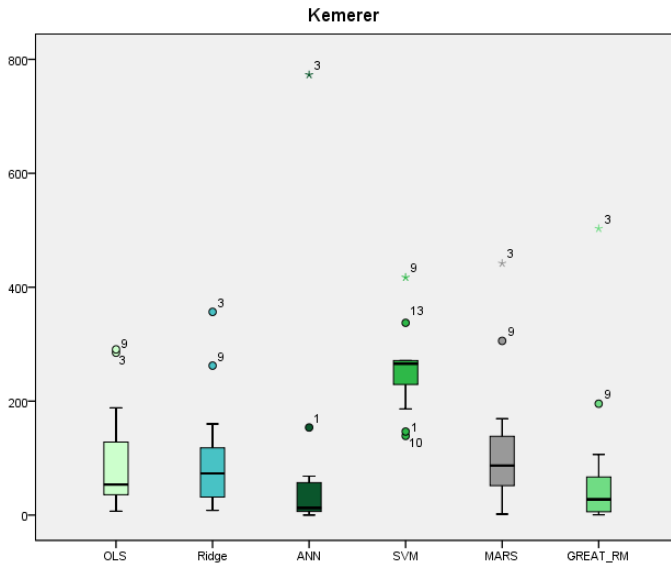


Fig. 14. Boxplot of Absolute Residuals for Kemerer Dataset

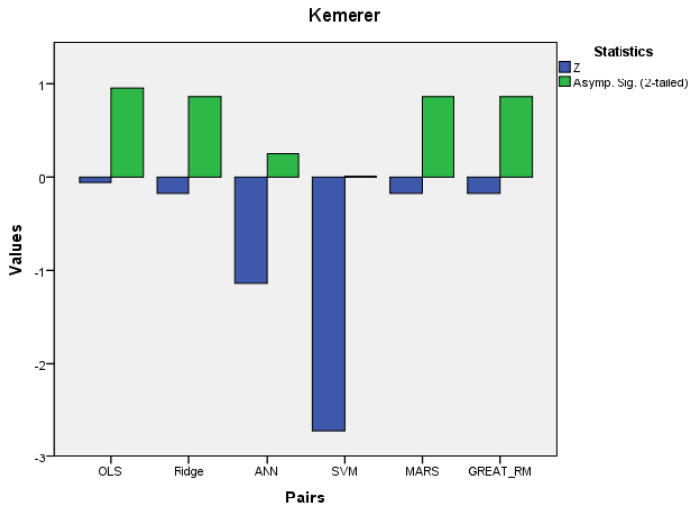


Fig. 15. Bar graph(WILCOXON Signed Rank Test)

- The medians for the *GREAT\_RM* techniques applied to the Kemerer dataset are very close to the median, as is clear from the values on the Y-axis, indicating that the estimates were closer to the minimum value.
- The range of absolute residuals of *GREAT\_RM* is smaller than the absolute residuals of others.
- On the other hand, the range of absolute residual values for SVM produced the worst individual estimations.

Table 13. Wilcoxon signed rank test (Test Statistics<sup>c</sup>)

Kemerer Dataset	OLS	Ridge	ANN	SVM	MARS	GREAT_RM
Z	-.419 <sup>a</sup>	-.551 <sup>a</sup>	-.419 <sup>a</sup>	-1.409 <sup>a</sup>	-.566 <sup>a</sup>	-.175 <sup>b</sup>
Asymp. Sig. (2-tailed)	.675	.582	.675	.159	.571	.861

a. Based on positive ranks. b. Based on negative ranks. c. Wilcoxon signed ranks test

*WILCOXON Signed Rank Test:*

The p-value in all of the cases is greater than 0.05, as shown in Table 13. Thus, the proposed methods are statistically significant.

### 5. COMPARISON OF GREAT\_RM WITH OTHER ESTIMATION METHODS

To ensure impartial assessment between our model and previous available results we have taken only those results for comparison that possess the same: (1) validation procedure, (2) number of projects, (3) number of features, and (4) treatment to missing values.

#### 5.1 Comparison for the Finnish Dataset

The best results have been achieved with the Finnish dataset, with an MMRE of 10.03% and a Pred (25) accuracy of 89.47%, with the Fair M-estimator. The lower edge of the boxplot for all cases overlaps the lower whisker. This shows that the data is probably skewed towards the lower end of the scale. The smaller sizes of the box with the robust estimator indicate a reduced variability of absolute residuals.

#### 5.2 Comparison for the Albrecht Dataset

On using the proposed methodology with the Albrecht dataset, the best MMRE = 24.16% and Pred (25) accuracy = 70.83% with the Huber M-estimator is obtained. This is an MMRE improvement of 36.09% over the GRACE technique [17], 26.94% over FGRA [23], and 1.94% over GRACE+[19]. The prediction accuracy has also been remarkable. There has been an improvement of 18.2% over GRACE, 20.83% over GRACE+ and an improvement of 42.23% over FGRA. In the study, carried out by Shepperd and Schofield [11], that made comparison between regression and analogy estimation models using the Albrecht dataset. The regression model resulted in MMRE=90% and Pred=33%, while analogy obtained MMRE=62% and Pred=33%. Thus, the results achieved by GREAT\_RM are significantly better than other estimation methodologies. The results of MdmRE and MMER obtained by the GREAT\_RM methodology are mentioned in Table 14.

Table 14. Comparison for the Albrecht Dataset

Albrecht Dataset				
	MMRE	Median MRE	Pred(25)	MMER
<b>GREAT_RM</b>	24.16	10	70.83	22.57
<b>GRACE+</b>	26.1	24.2	50	
<b>FGRA</b>	51.1	48	28.6	60.4
<b>GRACE</b>	60.25	21.35	52.63	

### 5.3 Comparison for the COCOMO-81 Dataset

GRACE [17] used GRA and stepwise regression on the COCOMO 81 dataset. The results obtained using GRA were MMRE=76.09% and Pred (25)= 20.63%. The results obtained based on regression were MMRE=1540.84% and Pred(25)=6.67%. Dolado [45] used regression and genetic programming and achieved MMRE=113% and Pred(25)=17% with regression, whereas, MMRE=23.2% and Pred(25)=15% was obtained using genetic programming. GRACE<sup>+</sup> [19] used GRA and obtained MMRE=49.8% and Pred(25)=29%, whereas, the best results of MMRE=23.2% and Pred(25)=66.7% have been achieved by Azzeh et al. on using Fuzzy GRA [23]. Using the proposed hybrid analogy technique the finest results were obtained on integrating GRA with stepwise regression. Thus, the result obtained demonstrate its applicability for the COCOMO-81 dataset, as it yielded better results than Song et al.[17, 19], Dolado [45], and Azzeh et al.[23]. The results obtained for MdMRE and MMER using the *GREAT\_RM* methodology are also mentioned in Table 15.

Table 15. Comparison for the COCOMO-81 Dataset

COCOMO 81				
	MMRE	Median MRE	Pred(25)	MMER
<b>GREAT_RM</b>	21.04	9.42	76.19	48.71
<b>GRACE+</b>	49.8	55.2	29	-
<b>FGRA</b>	23.2	14.8	66.7	25.6
<b>GRACE</b>	76.09	60.52	20.63	-

### 5.4 Comparison for the Desharnais Dataset

The Desharnais dataset has been extensively used for testing models based on software estimations. GRACE by Song et al. [17] obtained an MMRE of 49.83% with a Pred (25) accuracy of 30% with the Desharnais dataset. Mair et al. [46] also used analogy and obtained MMRE=57%. Shepperd [11] on using the Desahrnais dataset achieved MMRE=64% and Pred (25)=36%. Mair and Shepperd [46], however, used the hold out strategy, whereas, Song et al. and Azzeh used the jackknifing technique for GRACE<sup>+</sup>[19] by Song et al. and obtained MMRE=49.83% and Pred(25)=30%. The finest results so far have been by Azzeh et al. [23] with using Fuzzy GRA and they obtained the lowest MMRE=30% and a Pred (25) as high as 64.7. The results obtained by using the *GREAT\_RM* techniques have been superior to all the results mentioned above with an MMRE as low as 16.78% and Pred (25) as high as 74.02% when using GRA with stepwise regression. The GRA, when applied with OLS, also produced good results, as shown in Table 1. Thus, the result obtained demonstrates its applicability for the Deshanais dataset, yielding better results than those by Song et al. [17] and Azzeh et al. [23].

Table 16. Comparison for the Desharnais Dataset

Desharnais Dataset				
	MMRE	Median MRE	Pred(25)	MMER
<b>GREAT_RM</b>	16.78	7.84	74.02	31.63
<b>GRACE+</b>	41.4	29.2	45.3	-
<b>FGRA</b>	30.6	17.5	64.7	34.4
<b>GRACE</b>	49.83	33.93	30	-

With the OLS method, applied to the GRA, the dataset obtained equally good results with an MMRE of 18.19% with a Pred (25) of 90.9% . This is an MMRE improvement of 31.64 % over GRACE, 12.41 over FGRA, and 23.21 over GRACE<sup>+</sup>. The results of MdMRE and MMR obtained by the *GREAT\_RM* methodology are also mentioned in Table 16.

### 5.5 Comparison for the Kemerer Dataset

For the Kemerer dataset, the FairM-estimator obtained the best results when used with GRA. MMRE of 29.63 % and a Pred (25) accuracy of 60 % is an MMRE improvement of 29.2% over GRACE [17], 6.57% over FGRA [23], and with a Pred (25) accuracy improvement of 33.33% over GRACE [17] and 7.1% over FGRA [23]. The results however didn't improve over GRACE<sup>+</sup>. This may be because of the very small size of the dataset and may also be due to the feature selection process and outlier detection performed by Song et al.[19].

Table 17. Comparison for the Kemerer Dataset

Kemerer Dataset				
	MMRE	Median MRE	Pred(25)	MMER
<b>GREAT_RM</b>	29.63	16.56	60	27.93
<b>GRACE+</b>	19.6	13.8	78.6	-
<b>FGRA</b>	36.2	33.2	52.9	34.3
<b>GRACE</b>	58.83	46.94	26.67	-

## 6. THREATS TO VALIDITY

There are two types of threats to validity. One is the threat to internal validity and the second is the threat to external validity. The external validity is more crucial with respect to the internal validity as they are related to the generalization ability of the predicted models. For five publicly available datasets it has been cleared that the new proposed model achieved by integrating GRA and regression has improved predictive power, which is better than with the other conventional estimation methods, but we have not analyzed how a smaller or larger dataset than those used in the study would yield consistent results. The results have been shown consistent for sample sizes ranging from 15 to 77 samples. We do not think that increasing the sample size beyond this range would show any inconsistent results. The threats to validity can however be reduced by conducting more studies across varied datasets.

## 7. CONCLUSION

Producing accurate software estimates has always been a challenge, where no one method has established itself to the fullest to consistently deliver an accurate estimate. Analogy based estimation is still one of the most extensively used methods in the industry. It is based on finding efforts for similar projects from the project repository.

The proposed hybrid *EbA* methodology has certainly improved the estimation process. The results obtained on applying the proposed methodology to five publically available datasets for software effort estimation have been presented in the previous section. The results obtained are compared to various types of linear regression models like OLS, ridge regression, etc., and to

nonlinear models like neural networks, support vector machines, and Multivariate Adaptive Regression Splines (MAR Splines), etc. The results obtained using *GREAT\_RM* are not only better but are also encouraging with a lower MMRE, MdMRE, MMR, and higher Pred (25) for five publicly available datasets. The results are also significant when compared to three well known estimation models of GRACE, FGRA, and GRACE<sup>+</sup>. The results based on the Wilcoxon signed rank test for residuals illustrates that most of the models using *GREAT\_RM* produced statistically accurate predictions as their medians are not statistically different from the hypothesized median.

The empirical evaluations have revealed that the *GREAT\_RM* techniques can certainly enhance the estimation process and hence can be used as an alternative technique for early stage software estimation where the data is uncertain. The results obtained are also finer over our previous results wherein the value of  $k$  was fixed for each reference project [48].

This methodology can further be explored on some other large datasets with resampling methods in order to further enhance the validity of the produced results. This methodology can be worked out using other robust regression techniques like S-estimator, Least Trimmed Square or MM-estimator etc. Attribute weighting can also be incorporated along with feature selection, in order to study the impact of individual features on the prediction accuracy.

## REFERENCES

- [1] J. L. Deng, "Control problems of grey system". *System and Control Letters*, Vol.1, 1982, pp.288-94.
- [2] J.Deng, "Introduction to Grey System theory", *The Journal of Grey System*, Vol.1, No.1, 1989, pp.1-24.
- [3] J.Deng, "Grey information space", *The Journal of Grey System* Vol.1, No.1, 1989, pp.103-117.
- [4] J. M. Jou, P. Y.Chen, and J. M.Sun, "The grey prediction search algorithm for block motion estimation". *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.9, No.6, 1999, pp.843-848.
- [5] S. L. Su, Y. C. Su, and J. F.Huang, "Grey-based power control for DS-CDMA cellular mobile systems". *IEEE Transactions on Vehicular Technology*, Vol.49, No.6,2000, pp.2081-2088.
- [6] B.C.Jiang, S. L.Tasi and C. C.Wang, "Machine vision-based gray relational theory applied to IC marking inspection". *IEEE Transactions on Semiconductor Manufacturing*, Vol.15, No.4, 2002, pp.531-539
- [7] R. C.Luo, T. M.Chen, and K. L. Su, "Target tracking using a hierarchical grey-fuzzy motion decision making method". *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol.31, No.3, 2001, pp.179-186.
- [8] Y. F.Wang, "On-demand forecasting of stock prices using a real-time predictor". *IEEE Transactions on Knowledge and Data Engineering*, Vol.15, No.4, 2003, pp.1033-1037.
- [9] S. J.Huang and C. L.Huang, "Control of an inverted pendulum using grey prediction model". *IEEE Transactions on Industry Applications*, Vol.36, No.2, 2000, pp.452-458.
- [10] T. Mukhopadhyay, S. Vicinanza and M. J. Prietula, "Examining the feasibility of a case-based reasoning model for software effort estimation", *MIS Quarterly*, Vol.16, No.2, 1992, pp.155-171.
- [11] M. J. Shepperd and C.Schofield, "Estimating Software Project Effort Using Analogies", *IEEE Transaction on Software Engineering*, Vol.23, 1997, pp.736-743.
- [12] L. Angelis, I. Stamelo, "A simulation tool for efficient analogy based cost estimation," *Empirical Software Engineering*, Vol.5, 2000, pp.35-68.
- [13] J.W. Keung, B. A. Kitchenham, D. R. Jeffery, "Analogy-X:Providing Statistical Inference to Analogy-Based Software Cost Estimation", *IEEE Transactions on Software Engineering*, Vol.34, No.4, 2008.
- [14] B. Baskales, B. Turhan, A. Bener, "Software effort estimation using machine learning methods," *22nd international symposium on Computer and information sciences*, 2007, pp.1-6.
- [15] A. Idri, A. Abran, T. M. Khoshgoftaar, "Estimating Software Project Effort by Analogy Based on Linguistic Values," *Eighth IEEE International Symposium on Software Metrics (METRICS'02)*, 2002, pp.21.

- [16] M. Azzeh, D. Neagu and P. I. Cowling, "Analogy-based software effort estimation using Fuzzy numbers". *Journal of Systems and Software*, Vol.84, No.2, 2011, pp.270-284 [doi: 10.1016/j.jss.2010.09.028]
- [17] Q.Song, M.Shepperd and C.Mair,"Using Grey Relational Analysis to Predict Software Effort with Small Data Sets". *Proceedings of the 11th International Symposium on Software Metrics (METRICS'05)*, 2005, pp.35-45.
- [18] C. J. Hsu and C. Y. Huang, "Comparison and Assessment of Improved Grey Relation Analysis for Software Development Effort Estimation," *Proceedings of the 3rd International Conference on Management of Innovation and Technology (ICMIT'06)*, 2006, pp.663-667.
- [19] Q.Song and M. J.Shepperd, "Predicting software project effort: A grey relational analysis based method". *Expert Syst. Appl.* Vol.38, No.6, 2011, pp.7302-7316. [ doi:10.1016/j.eswa.2010.12.005]
- [20] S. J. Huang, N. H. Chiu and L.W. Chen, "Integration of the grey relational analysis with genetic algorithm for software effort estimation". *European Journal of operational and research* Vol.188, 2007, pp.898-909. [doi:10.1145/1540438.1540440]
- [21] M. V. Kosti, N. Mittas, L. Angelis, " DD-EbA: An algorithm for determining the number of neighbors in cost estimation by analogy using distance distributions", *3d Artificial Intelligence Techniques in Software Engineering Workshop*, 7 October, 2010, Larnaca, Cyprus.
- [22] G. Li, J.Ruhe, A. Al-Emran and M.M.Richter, "A flexible method for software effort estimation by analogy", *Empirical Software Engineering*, Vol.12, No.65, 2007, pp.106. [doi:10.1007/s10664-006-7552-4]
- [23] M. Azzeh, D. Neagu and P. I. Cowling, "Fuzzy grey relational analysis for software effort estimation", *Journal of Empirical software Engineering*, Vol.15, No.1, 2010. [ doi:10.1007/s10664-009-9113-0]
- [24] K. Srinivasan and D. Fisher, "Machine Learning Approaches to Estimating Software Development Effort," *IEEE Trans. Software Eng.*, Vol.21, No.2, 1995, pp.126-137.
- [25] G. Wittig and G. Finnie, "Estimating Software Development Effort with Connectionist Models," *Information and Software Technology*, Vol.39, No.7, 1997, pp.469-476.
- [26] C. Burgess and M. Lefley, "Can Genetic Programming Improve Software Effort Estimation? A Comparative Evaluation," *Information and Software Technology*, Vol.43, 2001, pp.863-873.
- [27] Y. Shan, R.J. McKay, C. J. Lokan and D.L. Essam, "Software Project Effort Estimation Using Genetic Programming", IEEE, Available at: <http://www.isbsg.org.au>, 2002.
- [28] A.Idri, A. Abran and T. M. Khoshgoftaar, "Estimating Software Project Effort by Analogy Based on Linguistic Values", Eighth IEEE International Symposium on Software Metrics (METRICS'02), 2002.
- [29] X. Huang, L. F. Capretz and J. Ren, "A Neuro Fuzzy Model for Software Cost Estimation", *Proceedings of the third International Conference on Quality Software (QSIC'03)* 0-7695 2015-4/03, IEEE, 2003.
- [30] Z. Chen, T. Menzies, D. Port, and B. Boehm, "Feature Subset Selection Can Improve Software Cost Estimation Accuracy," *ACM SIGSOFT Software Eng. Notes*, Vol.30, No.4, 2005, pp.1-6
- [31] P. Sentas, L. Angelis, I. Stamelos, and G. Bleris, "Software Productivity and Effort Prediction with Ordinal Regression," *Information and Software Technology*, Vol.47, 2005, pp.17-29.
- [32] A. F. Sheta, "Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects", *Journal of Computer Science* 2 (2): 118-123, ISSN 1549-3636, 2006.
- [33] M. Auer, A. Trendowicz, B. Graser, E. Haunschmid, and S. Biffl,"Optimal Project Feature Selection Weights in Analogy-Based Cost Estimation: Improvement and Limitations," *IEEE Trans.Software Eng.*, Vol.32, No.2, 2006, pp.83-92.
- [34] N.-H. Chiu and S.-J. Huang, "The Adjusted Analogy-Based Software Effort Estimation Based on Similarity Distances," *The J. Systems and Software*, Vol.80, 2007, pp.628-640.
- [35] K. Chaudhary, "GA Based Optimization of Software Development Effort Estimation", *GA Based Optimization of Software Effort Estimation, IJCSI*, Vol.1, 2010, pp.38-40.
- [36] M. Azzeh, D. Neagu and P. Cowling, "Improving Analogy Software Effort Estimation using Fuzzy Feature Subset Selection Algorithm", *PROMISE'08*, Leipzig, Germany, 2008.
- [37] V. Ch, M.K. Hari, T. S. Sethi, B. S. S. Kaushal and A. Sharma, "CPN-A Hybrid Model for Software Cost Estimation", 978-1-4244-9477-4/11, IEEE, 2011.

- [38] P. J. Huber, "Robust Estimation of a Location Parameter". *Annals of Mathematical Statistics*, Vol.35, 1964, pp.73-101.
- [39] P. J. Huber, Robust regression: Asymptotics, conjectures and Monte Carlo, *The Annals of Statistics*, Vol.1, 1981, pp.799-821.
- [40] D.F.Andrews, P.J. Bickel, F. R. Hampel, P.J. Huber, W.H. Rogers and J. W.Tukey, *Robust Estimates of Location: Survey and Advances*. Princeton University Press, Princeton, New Jersey,1972.
- [41] MATLAB® Documentation, <http://www.mathworks.com/help/techdoc/>
- [42] G.A.N.Mbamalu and M.E.El. Hawary, "Load Forecasting via Suboptimal Seasonal Autoregressive models and Iteratively Reweighted Least Squares Estimation" *IEEE Transactions on Power Systems*,Vol.8, No.1, 1993, pp.343-347.
- [43] V.Verardi, and C. Croux, "Robust regression in Stata", *Stata Journal*, StataCorp LP, Vol.9, No.3, 2009, pp.439-453.
- [44] PROMISE Repository of empirical software engineering data <http://promisedata.org/> repository
- [45] Dolado JJ (2001) "On the problem of the software cost function". *Journal of Information and Software Technology*, Vol.43, pp.61-72.
- [46] Mair C, Kadoda G, Lefley M, Phalp K, Schofield C, Shepperd M, Webster S , "An investigation of machine learning based prediction systems". *J Syst Software*, Vol.53, 2000, pp.23-29.
- [47] G. Nagpal, M. Uddin and A. Kaur, "A hybrid technique using Grey Relational analysis and Regression for Software Effort Estimation using Feature Selection" *International Journal of Soft Computing and Engineering (IJSCE)*, Vol.1, No.6, 2012.



### **Geeta Nagpal**

Geeta Nagpal, pursuing Ph D programme in the Department of Computer Science and Engineering, National Institute of Technology, Jalandhar, INDIA. She completed her Master's degree in Computer Science from Punjab Agricultural University, Ludhiana. She is presently working as Associate Professor in the Department of Computer Science and Engineering at National Institute of Technology, Jalandhar. Her research interests are Software Engineering, Databases and Data mining.



### **Prof. Moin Uddin**

Prof. Moin Uddin, Pro Vice Chancellor, Delhi Technological University, Delhi , INDIA. He obtained his B.Sc. Engineering and M.Sc. Engineering (Electrical) from AMU, Aligarh in 1972 and 1978 respectively. He obtained his Ph. D degree from University of Roorkee, Roorkee in 1994. Before joining as the Pro Vice Chancellor of Delhi Technological University, he was the Director of NIT, Jalandhar for five years. He has worked as Head Electrical Engineering Department and Dean Faculty of Engineering and Technology at Jamia Millia Islamia (Central University) NewDelhi. He supervised 14 Ph. D thesis and more than 30 M.Tech dissertations. He has published more than 40 research papers in reputed journals and conferences. Prof. Moin Uddin holds membership of many professional bodies. He is a Senior Member of IEEE.



**Dr. Arvinder Kaur**

Dr. Arvinder Kaur, Associate Professor, University School of IT, Guru Gobind Singh Indraprastha University, Delhi, India. She completed her masters degree in Computer Science from Thapar Institute of Engineering and Technology and Ph D from Guru Gobind Singh Indraprastha University, Delhi. Prior to joining the school, she worked with Dr. B. R. Ambedkar Regional Engineering College, Jalandhar and Thapar Institute of Engineering and Technology. Her research interests include Software Engineering, Object-Oriented Software Engineering, Software Metrics, Microprocessors, Operating Systems, Artificial Intelligence, and Computer networks. She is a lifetime member of ISTE and CSI. She is also a member of ACM. She has published 45 research papers in national and international journals and conferences. Her paper titled, "Analysis of object oriented Metrics" was published as a chapter in the book Innovations in Software Measurement (Shaker-Verlag, Aachen 2005).