IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Software Effort Estimation with Data Mining Techniques- A Review

Mohita Sharma[*1], Neha Fotedar[2]

[*1,2] Lovely Faculty of Technology and Science, Lovely Professional University, Phagwara, India

sharmamohita@yahoo.com

Abstract

Effort Estimation is an important task in cost prediction of the software. This task comes under the planning phase of software project management. In this paper, a review of different data mining techniques used for effort estimation has been presented. The techniques taken into consideration are Clustered techniques (K-Means, K-NN-K-Nearest Neighbour), Regression techniques (MARS- Multivariate analysis for regression splines, OLS - Ordinary least square regression, SVR-Support vector regression, CART- classification and regression trees) and classification techniques (SVM-Support vector machine, CBR-Case based reasoning). We can use the hybrid approach of these techniques for improving effort estimation.

Keywords: Support vector machine; constructive Cost Model; K-Means; person- month; data mining.

Introduction

Software effort estimation is one of the most important field in the software Engineering. Effort estimations are determined during the planning stage of the project. It provides the basis for subsequent planning, control, and decision making. In this paper, review of some of the basic effort estimation techniques has been done. After this, review of some popular data mining techniques used in software effort estimation have been presented. Advantages and disadvantages of each technique presented are also discussed.

Software Effort Estimation

Effort estimation is prediction of percentage and number of hours for the effort invested during a software project. Estimating the effort is very necessary and most analysed variable in recent years. It is used basically in project management. Software engineers were facing problem of effort predictions since 1950. Estimation overrun was occurring even for small projects. At early time, the effort estimation was based on regression analysis and mathematical formulae. SLIM- Software Life Cycle Management and COCOMO- Constructive Cost Estimation are the basic models for effort estimation. Tremendous growth of software system trade resulted in new technologies. In every field Software effort estimation requires additional concentration. Actual estimation is often a difficult task. Effort estimation techniques are generally classified into algorithmic and non-algorithmic techniques. Association in the algorithmic model provides a mathematical equation for estimation that is predicated upon the analysis of information gathered from antecedent developed comes. Non-algorithmic techniques support new approaches, like soft computing [21] techniques. The most tasks for software system development estimation are to determine the effort, cost and time of developing the project into consideration. So, correct effort estimation results in effective management of your time and budget throughout software system development. The estimation approaches for effort estimation are regression, analogy, expert judgment, work breakdown, function point, simulation, neural network, bayesian and combination of estimates. For effort estimation one can work on estimation methods, production functions, size measures, organizational issues, effort uncertainty assessments, measure of estimation performance and data set properties. If at the planning stage, developer has a good estimation of the factors, which will influence the cost then it will be smooth for developer in future to develop the project.

Data Mining Techniques

Data mining techniques are used in a variety of fields today. It has been applied in businesses for marketing and CRM-Customer Relationship Management. The improvements have been made in the data mining algorithms for using them in Software Engineering. For effort estimation, a great change has been made in the data mining algorithms. These changes have been made to increase the accuracy of software effort estimation. Many of the data mining techniques like OLS- Ordinary Least

Square, LSSVM- Least Square –Support vector machine [28], MARS- Multivariate Adaptive Regression Splines, LMS- Least Median of Square Regression, K-Nearest Neighbor, K-Means, Bagging- Bootstrap Aggregation, CBR- Case based Reasoning, CART-Classification and Regression tree are used in software effort estimation.

Literature Review

Software Effort Estimation (SEE) initially appeared in the fifties. Since then it has continually drawn attention in software package community specialists. It aims to develop, helpful models that constructively make a case for the development Lifecycle. SEE redirect the price of developing software systems.

Basic Methods

In 1960s Expert Judgment methods were developed. An algorithmic model Putnam (1970) developed a model for effort estimation based on Rayleigh curves known as SLIM (Software Life Cycle Management) [25]. Developer Barry Boehm (1981) developed COCOMO as a constructive cost model [3]. An easy going & understandable model advanced by Barry Boehm could call the effort & time period of the project. This model is a bridge on input relating to the size of the resulting system. The COCOMO model calculates the effort by: E = c (KLOC) ^{d} where,

E is estimate effort in man month and c, d are the constants. After the development of COCOMO, Barry Boehm later on developed COCOMO 2.0, to overcome problems and misses those were found in the first version of the COCOMO [4]. Howard Rubin designed ESTIMAC model to estimate effort at the conception stage. ESTIMAC behaves as a closed model as the way ESTIMACS translates the input to the effort was not clear. Rubin identified the six critical estimation dimensions: effort hour, staff size, cost, hardware resource requirement, risk in development and portfolio impact [15]. Allan Albrecht developed measurement method called function point at IBM. For LOC (line of code) techniques many problems were faced as: lack of universally accepted definition for exactly what line code really is. Other side line of code is language dependence [1]. Function point defines the complexity of software system in terms of functions that system is delivered to the user. It includes combination of five basic software components (input, output, master files, interfaces, inquiries). The values of software components can be low, average, or high. Krishnamoorthy, F. Douglas Fisher Srinivasan (1995) applied the machine learning approach for software effort estimation. In this paper the Back propagation algorithm on COCOMO dataset has applied. Three experiments on different datasets are being performed. They concluded that Back propagation is competitive against traditional approaches but quite sensitive [11]. The effort estimation was improved with the help of different data mining techniques. In Fig. 1. some of the data mining techniques used in effort estimation are depicted .

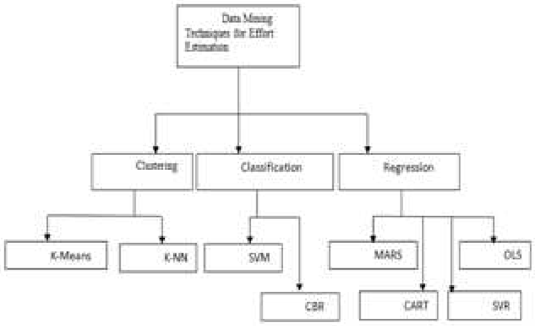

Figure 1 Data mining techniques for effort estimation

Ordinary Least Square Regression

Ordinary Least Squares regression is the oldest and most generally applied technique for software system effort estimation. This well-documented technique fits a linear regression function to a knowledge set containing a dependent, EI, and multiple freelance variables, x_i (1) to x_i (n) this kind of regression is additionally unremarkably stated as multiple regression. OLS regression assumes the subsequent linear model of the data by (1)

$$e_i = x'_i \beta + b_0 + €_i \qquad (1)$$

where x’ represents the row vector containing the values of the ith observation, $x_i(1)$ to $x_i(n)$. β is the column vector containing the slope parameters that are estimated by the regression, and b_0 is the intercept scalar. This intercept can also be included in the β vector by introducing an extra variable with a value of one for each observation. $€_i$ is the error associated with each observation [6]. Myrtveit, Ingunn, and Erik Stensrud (1999) estimated software project effort using OLS regression and Case based reasoning. They have worked on COTS-Commercial of the shelf data set. In this the comparison between machine learning and regression techniques has performed. The comparison is performed with the help of a data sample, an accurate indicator, and cross validation with reliability parameters [13]. Kevin Strike, Khaled El Emam, and Nazim Madhavji (2001) had done the study on the missing values in the field of software effort estimation using OLS regression. By this it is found that all the missing data techniques perform well with small biases and high precision [16]. Tim Zhihao Chen, Jairus Hihn, and Karen Lum (2006) done the research on deviations exhibited by different techniques of software effort estimation using OLS regression. A COSEEKMO tool is being developed for effort estimation. This tool uses standard t-tests [24]. The advantage of using COSEEKMO is that it is fully automatic. This tool is used for selecting the alternative method. The problem of using is that OLS has a restriction on the input process data.

Bagging

Bagging (Bootstrap Aggregation) is a machine learning algorithm. It is used for accuracy of machine learning algorithms used in statistical classification and regression [14]. Petrônio Braga, Adriano LI Oliveira, Gustavo HT Ribeiro, and Silvio RL Meira (2007) have worked on the improvement of various data mining techniques like regression trees, modal trees, Multilayer perceptron, linear regression, and support vector regression for effort estimation. SVR is a stable algorithm and they were not able to improve the SVR for NASA datasets. Bagging is used as the averaging of regression problems and prediction process can be improved with the help of bagging [6]. The disadvantage of bagging is Complexity. [16].

K-Nearest Neighbor

The k-nearest neighbor algorithm is one of the machine learning algorithms. In this an object is assessed by a majority vote of its neighbors, with the element being appointed for the category commonly amongst its k nearest neighbors (k may be a positive number, generally small). If k = 1, then the article is solely appointed for the category of that single nearest neighbor. Yigit Kultur, Burak Turhan, and Ayse Bener (2009) provides a technique which used ensemble based neural networks. They generate a combined approach of ANN and K- nearest neighbor [32]. By the combination of this the efficiency of

Multivariate Adaptive Regression Splines

MARS is a novel technique introduced by Milton Friedman. MARS is a nonlinear and the statistic regression technique exhibiting some attention-grabbing properties like easy interpretability, capability of modeling complicated nonlinear relationships, and quick model construction. It conjointly excels at capturing interactions between variables and so could be a promising technique to be applied within the domain of effort prediction. [6]. MARS fits the data as depicted by (2) .

$$e_i = b_0 + \sum_{k=1}^{K} b_k \prod_{l=1}^{L} h_l(x_i(j)) \qquad (2)$$

Where b_0 and b_k are the intercept and the slope parameter, respectively. h_l $(x_i(j))$are called hinge functions and are of the form max(0, xi(j)-b) in which b is called a knot.[6]. Geeta Sikka, Arvinder Kaur, and Moin Uddin (2010) have worked on the comparison of different data mining algorithms like multivariate adaptive regression (MAR), support vector machine (SVM), k-nearest neighbor (kNN) for calculating estimation based on function points. In this the repository from IFPUG (International Function point user group) has chosen. In this paper for finding the effort work is done on missing values [27]. The conclusion is drawn that MAR gives lowest mean relative error. SVM and ANN- artificial neural network are also good for function point analysis.

Support Vector Machine- SVM

SVM is introduced in COLT-92 by Boser, Guyon & Vapnik. It is theoretically well motivated algorithm. It is developed from statistical learning theory by Vapnik & Chervonenkis since the 1960s.

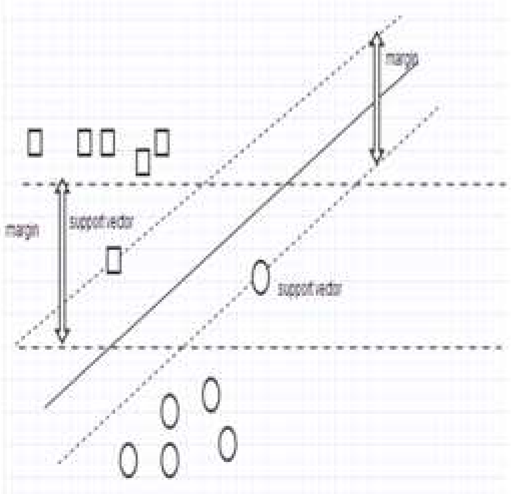

Figure 2. SVM Margin [2]

Fig. 2 shows an example of SVM margin where boxes and circles are different kinds of elements beyond the support vectors. According to the support vectors the attributes are being divided into two parts. No any attribute will be in the margin.

In SVM data is being separated into training and testing sets. Each instance in the training set contains one value that is known as target value or class label and contains several attributes known as observed variables. SVM finds a linear separating hyperplane. SVMs are a new promising non-linear, non-parametric classification technique. It is used in many fields like data mining, bioinformatics, artificial intelligence, software engineering, text etc. SVM is used in binary classification tasks, in the medical diagnostics, optical character recognition, electric load forecasting and other fields. SVM can be used in increasing the efficiency of effort estimation with the combination of K-Means. Amanjot and Raminder (2012) have worked on the survey of

Support vector machine (SVM) and K-NN to find the efficiency in an effort. SVM was found to be better than K-NN as SVM training is relatively easy. S.M. Mousavi, Seyed Hossein Iranmanesh (2011) have used the LS-SVM and Genetic Algorithm for effort estimation [18]. LS- SVM is used for supervised learning and genetic algorithm is used for the optimization of the parameters.

Classification and Regression Tree

CART is a non parametric algorithm and does not require functional parameters. CART can handle outliers very easily. CART generate binary tree until the final result obtains [6]. Shepperd and Schofield (1997) described the use of analogies for estimating software project effort using OLS regression and CART. Estimation by analogy is able to operate in those areas, where it is not possible to create an algorithmic model. But, the main problem with analogy systems is that it requires considerable amount of computation. Estimation has done on the data set from DPS database [23]. It is hard to implement CART in practical life because of its complexity and unstable samples [31].

CBR- Case Based Reasoning

CBR is a technique for managing and victimization information that may be organized as a separate abstraction of events or entities that are unit restricted in time and area. Every such abstraction is termed a case. It searches for the foremost similar cases and the effort is also calculated by these retrieved cases. This system is often utilized in software system effort estimation [26]. K. Gayathiri, Dr. T. Nalini, Dr. V. Khanaa(2013) has done the study of various data mining techniques applied on effort estimation like ordinary least square regression, pace regression, case based reasoning. The calculations are performed on COCOMO datasets. A good choice of attributes is needed to have less effort and optimized cost [17].

K- Means

The k-means technique can turn out specifically k completely different clusters of greatest potential distinction. K is positive range. The steps for k-means clustering from start to end are described in fig. 2. The grouping is completed by minimizing the total of squares of distances between centroid and data. Thus, the aim of K-means cluster is to classify the information into clusters [13]. Nazish Murtaza, Ahsan Raza Sattar, and Tasleem Mustafa (2010) used K-Means data mining technique with NN- Neural Network. They have worked on water supply in agriculture field to overcome the problem of wrong estimation of cost for the use of water. A comparison of the K-Means and My K-means algorithm for handling the outliers has been conducted. Least square (statistical technique) and neural networks (machine learning) estimation method for training the data are being used. Neural networks and My K-Means give more accuracy for effort estimation for water management [13].

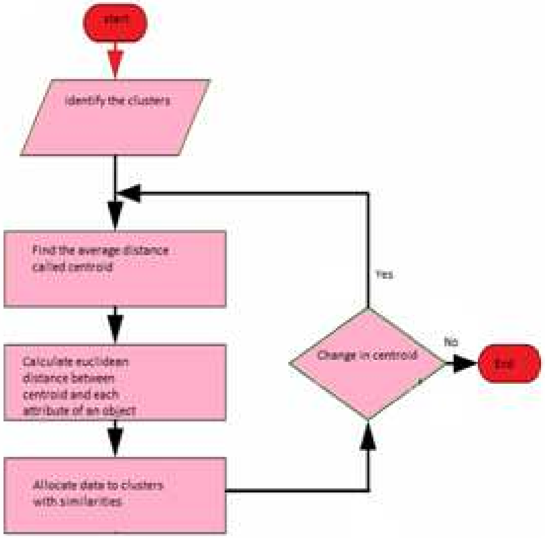
start
identify the clusters
Find the average distance called centroid
Calculate euclidean distance between centroid and each attribute of an object
Allocate data to clusters with similarities
Change in centroid
Yes
No
End

Figure 3 Steps for K-Means

Omer Faruk Sarac and Nevcihan Duru (2013) have used ANN- artificial neural network and k-means for effort estimation. The estimation is performed with COCOMO data set. Output from ANN will be the input for K- Means. By the combination of ANN and K-Means a model is developed, which is stable for the MMRE and MRE calculations [29]. Hari, C. V. M. K., T. S. Sethi, B. S. S. Kaushal, and A. Sharma. (2011) developed a hybrid technique which is the combination of PSO - Particle Swarm Optimization technique, k-means of data mining and back propagation technique of neural network. In this K-mean is used to cluster the data which is non linear. PSO is used for the selection of random data values for optimal values and then back propagation technique is used for training the data. The MMRE for this hybrid approach is 34.9 [12]. COCOMO data sets has chosen for this [12].

If an object has two attributes x1 and y1, and centroid of the cluster is x2 and y2, then distance is calculated by:

$$\text{Distance} = \sqrt{(x\ 2\text{-}x1)^{2} + (y2\text{-}y1)^{2}}$$

Support Vector Regression-SVR

SVR is used to solve the matter of a distributed solution in ridge regression. In SVR springs are attached to the tubes as compared to the ridge regression where springs are attached between data cases and decision surface [22]. Anna Corazza, et al. (2010) uses Support Vector Regression and tabu search for showing the more efficient effort estimation results. They have used datasets from PROMISE repository and from tukutuku database [7]. Jin-Cherng Lin, Yueh-Ting Lin, Han-Yuan Tzeng, and Yan-Chin Wang (2011) introduced a model which combines genetic algorithm with support vector regression. In this paper, chromosomes and crossover are used. Chromosomes and crossover are the part of genetic algorithm. Initialization parameter which is helpful for defining the solution is given in the form of chromosomes. Crossover is used in this, to combine the parameters from the different chromosomes with multiple methods. The methods can be single crossovers, two crossovers, uniform crossovers and arithmetic crossovers. Then mutation technique is applied on the combination. Then SVR predicts model is applied to the training data. Then calculation is done form the fit value by test data. The loop is being continued to check the reach generation. If a generation is reached, then choose the best model. With test data SVR predict is done, get the predict value and end. SVR is used for selection of the best model and for prediction of the value. Thus the combination of Genetic algorithm and SVR give more efficiency [7]. The MMRE value of this hybrid approach is 0.2085. Here the testing and verifiability is performed using historical data in COCOMO datasets, Desharnais datasets, Kemerer datasets and Albrecht datasets. Prediction level and mean magnitude of relative error are used to show the estimation. Sweta and Shashankar (2013) have done a comparative study of COCOMO, MOPSO - Multiple objective particle swarm optimization and support vector regression. The work is performed on accuracy and error rate. It is observed that SVR gives better result as compared to COCOMO and MOPSO [19].

Summary of Methods and Data Mining Techniques used for Effort Estimation

In table I the summary of different data mining techniques has been given with advantages, disadvantages and with MMRE - Mean Magnitude of Relative Error value of the techniques for effort estimation performed on various datasets. MMRE can be calculated by the following formula of calculation of the mean of MRE-magnitude of relative error. [6]

$$MRE= \frac{|\text{Actual Effort}_i - \text{Predicted Effort}_i|}{\text{Actual Effort}_i}$$

Where i is observation, whose effort we will have to calculate.

Table I

S. No.	**Software Effort Estimation Techniques**					
	Technique	**Key Idea**	**Advantage**	**Disadvantage**	**MMRE**	**Datasets**
1.	Expert Judgement [4]	Based on the judgement of experience of the experts [2].	Simple to understand [20].	It will be helpful only if new software is similar to earlier software	0.71[20]	Samples of bank data
2.	SLIM [4] [25]	Use Rayleigh function.	Helpful for saving the time	It has a great dependency on source lines of code.	7.72[15]	Business data sets
3.	COCOMO [15][4]	Effort and cost are predicted based on the size of the software.	Easily adjusted according to needs of the organization.	Should have proper knowledge about the size of the project.	0.52[4]	COCOMO81
4.	OLS [6]	Based on fits linear regression function.	It is simple method and easy to understand.	An attribute is removed, if more than 25% of the attribute values are missing. It Cannot handle highly correlated values.	0.37[13]	COTS project

5.	Bagging [6]	Used in statistical classes and regression. [14]	It can run in parallel mode. It can handle unstable classifiers in a better way [14]	Lack of interpretation as it is a linear combination of decision trees. [6]	0.2103[6] NASA
6.	K-NN [32]	Use voting of neighbours	It is simple so used for recognition problems well.	It is a lazy learner as there is no need to train the data.	-0.003[27] International Function Point User Group
7.	MARS [8]	Works on non-linear relationships [8]	It is used for capturing communication between variables [8].	It has low dimensionality because of nonparametric smoothers. [10]	Not known
8.	SVM [18]	A non-linear machine learning technique based on classification and regression. [9]	SVM is less overfitting and optimally separate the data[9]	There is problem of choosing kernals. There are discrete data obtained by which more problems can be created. [18]	0.0999[27] International Function Point User Group
9.	CBR [8]	Analogy cases are made and used.	It is easily understandable [9] and Useful where domain is difficult to model. Potential to lessen the problem of outliers.	A complex computation is required [Shepperd]	0.07[13] 48 industrial COTS project.
10.	CART	Tree based approach	It is simple to use and can easily handle complex situations.	Unstable samples	0.569[20] Samples of bank data
10.	K-Means [13]	Based on clustering of data by distance between centroid and data. [13]	Very fast computation and simple to understand.	It is applicable only when mean is defined. Number of clusters should be known in advance.	0.3067[30] Data on agriculture in Pakistan
11.	SVR [5].	Based on structural risk minimization principle [5] [19].	Helpful to overcome the matter of a distributed solution in ridge regression [5] [19].	Difficult to handle discrete data [5] [19].	0.2085[22] Combined with Genetic Algorithm COCOMO81

Conclusion

In this paper some of the data mining techniques have been elaborated to improve the accuracy of software effort estimation. Effort has been calculated on the basis of MMRE value. The technique which gives less MMRE value is assumed to be better. In future, hybrid approach of any of the data mining techniques for increasing the accuracy in effort estimation can be used. One can take datasets like NASA- National Aeronautics and Space Administration, COCOMO81, IFPUG. Some authors have used datasets from COTS projects also.

References

[1] Albrecht , A. J., & Gaffney Jr, J. E. Software function, source lines of code, and development effort prediction: a software science validation. Software Engineering, IEEE Transactions on, (6), 1983, pp. 639-648

[2] Auria, L., & Moro, R. A. Support Vector Machines (SVM) as a technique for solvency analysis. 2008.

[3] Boehm, B.W. Software Engineering Economics. Prentice-Hall, Englewood Cliffs, N.J. 1981.

[4] Boehm, Barry, Chris Abts, and Sunita Chulani. "Software development cost estimation approaches—A survey." Annals of Software Engineering 10, no. 1, 2000, pp. 177-205.

[5] B. Debasish, P. Srimanta and P. Dipak. "Support vector regression." Neural Information Processing-Letters and Reviews, 2007, pp. 203-224.

[6] B. Petronio L, O. Adriano L.I. , H.Y Gustavo, R. Ribeiro and Silvio , " Bagging predictors for estimation of software project effort," Neural Networks, IJCNN. International Joint Conference on IEEE, Aug. 2007, pp. 1595-1600.

[7] Corazza, Anna, M. D. Sergio DI, F. Filomena , G. Carmine , S. Federica, and M. Emilia. "Using tabu search to configure support vector regression for effort estimation." Empirical Software Engineering. 2013, pp. 1-41.

[8] D. Karel, V. Wouter , M. David, and B. Bart "Data Mining Techniques for Software Effort Estimation: A Comparative Study" IEEE transactions on software engineering, vol. 38, no. 2. pp. march/april 2012, pp. 375-397.

[9] D. Liuhuan, Y. Yuan, and C. Yudong. "Using Bagging classifier to predict protein domain structural class." Journal of biomolecular structure & dynamics 24, no. 3, 2006, pp. 239-243.

[10] F. John, Jerome. "Multivariate adaptive regression splines." The annals of statistics , 1991, pp. 1-67.

[11] F. Douglas Fisher Srinivasan, Krishnamoorthy, "Machine learning approaches to estimating software development effort." Software Engineering, IEEE Transactions on 21, no. 2, 1995, pp. 126-137.

[12] Hari, C. V. M. K., S. Tegjyot, K. B.S.S., S. Abhishek. "CPN-a hybrid model for software cost estimation."Recent Advances in Intelligent Computational Systems (RAICS), 2011. Sep. 2011, pp. 902-906.

[13] I. Myrtveit, E. Stensrud, and M. Shepperd, "A Controlled Experiment to assess the benefits of Estimation with analogy and Regression Models," IEEE Trans. Software Eng., Aug.1999, pp. 510-525.

[14] K. Machová, ,B. František, and B. Peter. "A bagging method using decision trees in the role of base classifiers." Acta Polytechnica Hungarica 3, no. 2. Pp. 121-132, 2006.

[15] Kemerer, Chris F. "An empirical validation of software cost estimation models."Communications of the ACM 30, no. 5, 1987: 416-429.

[16] K. Strike, K.E. Emam, and N. Madhavji, "Software Cost Estimation with Incomplete Data," IEEE Trans. Software Eng., vol. 27, no. 10. Oct. 2001 pp. 890-908.

[17] K. Gayathiri , T. Nalini, and V. Khanna. "Data mining techniques for software effort estimation to improve cost efficiency." International Journal Of Engineering And Computer Science ISSN: 2319-7242 Volume 2 Issue 4. April, 201, pp. 1215-1220.

[18] Klair, S. Amanjot Singh, and K. Raminder Preet. "Software Effort Estimation using SVM and kNN." International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012). 2012, pp. 146-147.

[19] K. Sweta and P. Shashankar,"Comparison and Analysis of Different Software Cost Estimation Method," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.1. 2013, pp. 153-157.

[20] L. Briand, K. E. Emam, D. Surmann, and I. Wieczorek, "An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques," Proc. 21st Int'l Conf. Software Eng., May 1999, pp. 313-323.

[21] L. Jin-Cherng, L. Yueh-Ting, T. Han-Yuan, and W. Yan-Chin. "Using Computing Intelligence Techniques to Estimate Software Effort." International Journal 4. 2013, pp. 43-53.

[22] L. Jin-Cherng Lin, C. Chu-Ting and H. Sheng-Yu, "Research on Software Effort Estimation Combined with Genetic Algorithm and Support Vector Regression." In Computer Science and Society (ISCCS),

2011 International Symposium on, 2011, pp. 349-352.

[23]M. Shepperd and C Schofield "Estimating Software Project Effort Using Analogies," IEEE Trans. Software Eng. Aug 1997, pp. 736-743.

[24]Menzies, Tim, C. Zhihao, H. Jairus, and Karen Lum. "Selecting best practices for effort estimation." Software Engineering, IEEE Transactions on 32, no. 11. 2006, pp. 883-895.

[25]Matson, Jack E., Bruce E. Barrett, and Joseph M. Mellichamp. "Software development cost estimation using function points." Software Engineering, IEEE Transactions on 20, no. 4, 1994, pp. 275-287.

[26]Patnaik, K. S., S. Malhotra, and Bibhudatta Sahoo. "Software Development Effort Estimation using CBR: A Review," 2004, pp. 1-13.

[27]S. Geeta, K. Arvinder, U. Moin," Estimating Function Points: Using Machine Learning and Regression Models,"International Conference on Education Technology and Computer (ICETC) , IEEE, 2010, pp. 3-52.

[28]S.M. Mousavi, I. Sayed Hossein, "Least Squares Support Vector Machines With Genetic Algorithm For Estimating Costs In NPD Projects," Communication Software and Networks(ICCSN) ,IEEE 3rd International Conference. May 2011, pp. 127-13.

[29]S.F. Omer, and D. Nevcihan. "A novel method for software effort estimation: Estimating with boundaries." Innovations in Intelligent Systems and Applications (INISTA), 2013 IEEE International Symposium on. IEEE, 2013, pp. 1-5.

[30]T. Nazish Mur ,T. Ahsan Raza Sat and T. Tas leem Mus, "Enhancing the Software Effort Estimation using Outlier Elimination Methods for Agriculture in Pakistan. Pakistan Journal of Life and Social science.2010. pp. 54-58.

[31]Timofeev, Roman. "Classification and regression trees (cart) theory and applications." Humboldt University, Berlin (2004).

[32]Y. Kultur, B. Turhan, A. Bener, "Ensemble of neural networks with associative memory (ENNA) for estimating software development costs", Knowledge-Based Systems, vol. 22, August 2009, pp. 395–402.