Machine learning methods with feature selection approach to estimate software services development effort

Amid Khatibi Bardsiri*

Computer Engineering Department,
Science and Research Branch,
Islamic Azad University, Tehran, Iran
Email: a.khatibi@srbiau.ac.ir
*Corresponding author

Seyyed Mohsen Hashemi

Computer Engineering Department,
Science and Research Branch,
Islamic Azad University, Tehran, Iran
Email: isrup@yahoo.com

Abstract: Estimate of the effort required for software services development has been a most important topic in the field of service in recent years. Exact estimate of effort is a key factor for project's successful management and control. Over and underestimation waste system resources endanger the position of the related company. The development effort estimation is done with the help of expert judgement, algorithmic and machine learning methods. Recently, several methods of machine learning have been used to estimation software services effort and look much better than the other two groups. This paper presents an experimental evaluation of the effectiveness of these methods with feature selection approach and done a thorough comparison of their accuracy. Evaluation and comparison have been made onto two famous datasets NASA and ISBSG and results are well demonstrated position of each one of these methods.

Keywords: effort estimation; software service; machine learning; comparison.

Reference to this paper should be made as follows: Bardsiri, A.K. and Hashemi, S.M. (2017) 'Machine learning methods with feature selection approach to estimate software services development effort', *Int. J. Services Sciences*, Vol. 6, No. 1, pp.26–37.

Biographical notes: Amid Khatibi Bardsiri received his BS in Computer Software Engineering from Shahid Bahonar University, Kerman, Iran in 2008, and his MS in Software Engineering from Islamic Azad University, Tehran, Iran, in 2010. He is currently undertaking his PhD in Science and Research Branch of Islamic Azad University, focusing on software metrics and measurement. He published about 25 research papers in international journals and conference proceedings. His areas of research include information systems engineering, software development, software metrics, grid computing and cloud computing.

Seyyed Mohsen Hashemi received his MS in Computer Science from Tehran Polytechnic University in 2003, and his PhD in Computer Science from the IAU University in 2009. He is currently the Dean of the Software Engineering and Artificial Intelligence Department, Science and Research Branch, IAU University, Tehran. He works on E-services & IBM SSME.

1 Introduction

On time and budget determined delivery of service is one of the main concerns of the most software companies. The necessary effort to develop a software service is among the most important and effective parameters of a project. Since the estimation process should be carried out in initial phases of the project, a reliable method is needed to be able to work with initial and little data (Jones, 2007). Different methods have been proposed to predict the effort which can be categorised in six groups: parametric methods such as SEER-SEM, COCOMO (Boehm and Valerdi, 2008), expert judgement such as WBS, Delphi methods (Jørgensen and Halkjelsvik, 2010), learn base models such as ABE (Phannachitta et al., 2013), regression methods such as OLS, ROR (Jeffery et al., 2001), dynamic models, and hybrid models (Dejaeger et al., 2012).

The introduction of function point (FP) by Albrecht and Gaffney (1983), was one of the important events in software measurement which gave the possibility of measuring the first levels of the project and prevented the negative effects of the previous method, LOC (Albrecht and Gaffney, 1983). Many changes in software development methodology and progress in estimation methods resulted in development of a new model called COCOMO II by Boehm et al. (2000). On the other hand, because of the inability of algorithmic methods in controlling dynamic behaviour of software projects and the lack of complete information of a project in primary stages, non-algorithmic methods have been presented.

Expert judgement method which was presented in 1963 is an example of these methods (Dalkey and Helmer, 1963). In this method, expert people share their ideas about the estimate value to achieve an agreement. CART, is another method of non-algorithmic method groups which attain the effort value in the leaves of the trees by making a tree and using the previous projects (Breiman et al., 1984). The most popular non-algorithmic estimation method is ABE method which was presented in 1997 (Shepperd and Schofield, 1997). This method uses comparison of a project with other similar historical cases. The comparison is based on the features of two projects. Moreover, other smart methods such as neural network, fuzzy rules and different methods of data mining have been used in effort estimation area (Azzeh et al., 2010; Dejaeger et al., 2012; Shukla et al., 2014).

In recent years, machine learning techniques have been used extensively in the field of estimating effort and have shown good performance (Srinivasan and Fisher, 1995; Dejaeger et al., 2012; Wen et al., 2012; Bardsiri and Hashemi, 2014). Despite the many improvements, yet are not well defined status of each of these methods and researchers are having difficulty in choosing them. The purpose of this article is the assessment and detailed comparison of different types of these methods.

This paper has been organised in five sections. The second section reviews the related works. Sections 3 describe machine learning methods. The empirical evaluation has been presented in Section 4, and Section 5 includes conclusions.

2 Related works

Many techniques have been introduced in the past years to estimate the required effort and cost for developing a software service. These methods have been initiated by simple equations and assumptions, and have now achieved complicated techniques (Bardsiri and Hashemi, 2014). These techniques can be divided into the three general groups below:

a Expert judgement: In this method, which was proposed in the late 1960s (Dalkey and Helmer, 1963) and is still widely employed in various software companies, domain experts are asked to give their opinions on the required effort. Various amounts are expressed and, typically, their median is returned as the final required effort. The Delphi method is an example of this class of techniques (Moløkken-Østvold and Jørgensen, 2004).

b Algorithmic models: These models, which use mathematical relations and equations, seek to discover a relationship between service attributes and the required effort, are usually suitable for specific cases, and are adjusted and calibrated depending on the existing conditions. COCOMO, SLIM, SEER-SEM are examples of this type of methods (Khatibi and Jawawi, 2011).

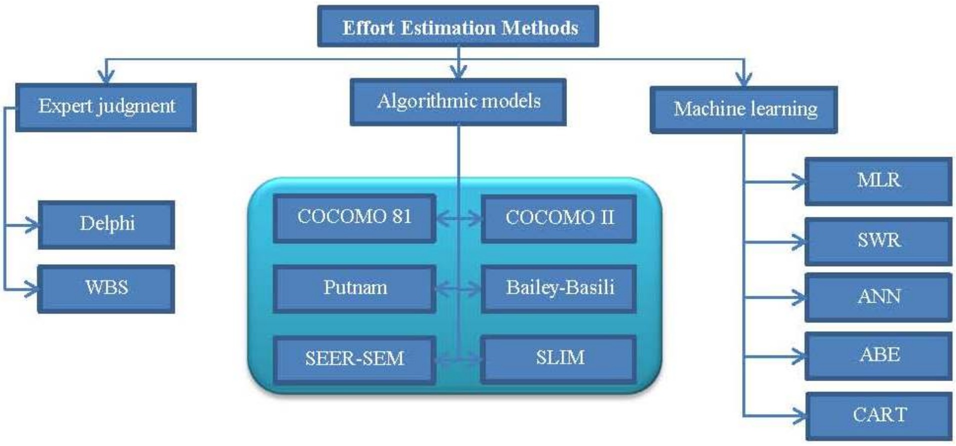
c Machine learning: These methods look to construct and study algorithms that can learn from datasets, are applied to inputs of the related problem, and help in the decision-making. Fuzzy theory, decision tree, ANN, and regression are examples of this class of methods (Wen et al., 2012).

Some of the benefits of machine learning methods include the ability to model complex relationships between dependent and independent variables and also power of learning from historical data. One of the disadvantages of the algorithmic methods, lack of flexibility and the need to calibrate themselves. These methods also do not have the ability to find the complex relationships between variables. Different kinds of regression (Dejaeger et al., 2012), COCOMO models and COCOMO II (Boehm, 1981; Boehm et al., 2000) are the most famous algorithmic models, and ABE (Shepperd and Schofield, 1997), CART (Breiman et al., 1984), Expert judgement (Dalkey and Helmer, 1963) and artificial neural network (Araújo et al., 2012), learning and artificial intelligence techniques (Azzeh, 2011), fuzzy rules and optimisation algorithms (Ahmed et al., 2005) are the most popular non-algorithmic methods. Figure 1 shows the different types of effort estimation methods and their subsets.

3 Machine learning methods

In this section, briefly be explained five different methods of the most important machine learning and continues to evaluate and compare these methods. It is important to note that nature of these models is different with each other completely (Dejaeger et al., 2012; Bardsiri and Hashemi, 2014).

Figure 1 Various types of effort estimation methods (see online version for colours)



3.1 SWR and MLR

Regression methods are among the oldest estimation methods and try to fit a function to a set of data. The dataset includes a dependent variable *E* and several independent variables X_i, and the linear equation (1) is considered for the data (Bardsiri et al., 2013a, 2013b, 2014):

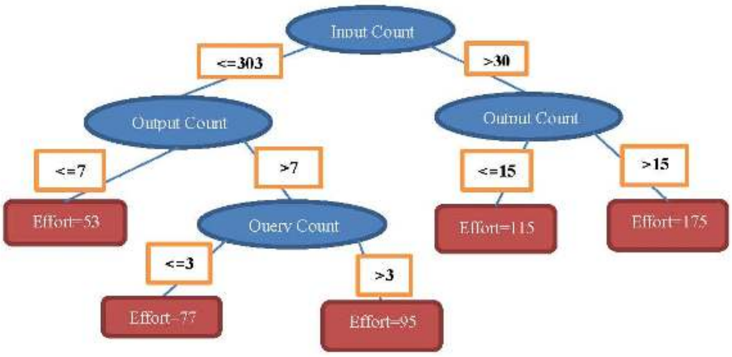
$$Y = B_1 X_1 + B_2 X_2 + \cdots + B_n X_n + b \tag{1}$$

In this equation, *B* is the slope of the line and *b* the value of the intercept, which can obviously be obtained by adding the one's column to the *X* vector In regression models, the purpose is to find the *B* and *b* coefficients in such a way that error is minimised. Multiple linear regression (MLR) and step wise regression (SWR) are examples regression models (Mendes et al., 2003).

3.2 Classification and regression trees

The purpose in classification and regression trees (CART) is to build a structured decision tree for classifying the set of instances in the dataset. The partition criterion is the simple testing of the features of the instances, and the tree is built recursively using simple if-then rules (Breiman et al., 1984). Each instance, depending on the values of its features, moves on the tree and reaches a specific leaf (which, here, is the amount of effort). This model was used in some of the previous studies (Dejaeger et al., 2012; Bardsiri et al., 2013a, 2013b; Benala et al., 2014; Zhang et al., 2015). The simple regression tree structure is presented in Figure 2.

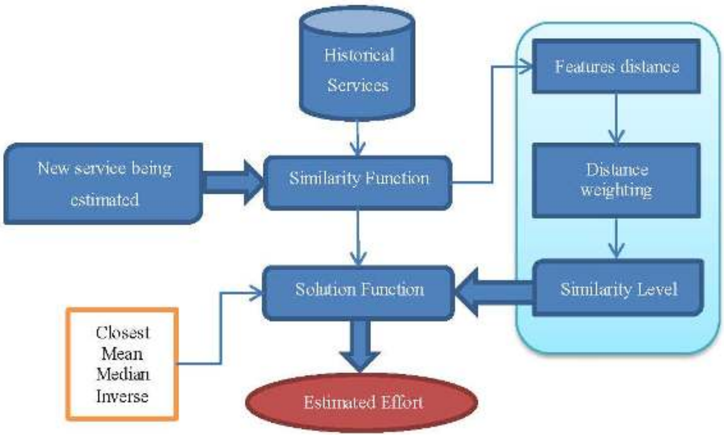
Figure 2 Example of a regression tree for software services effort estimation (see online version for colours)



3.3 *Analogy-based estimation*

The analogy-based estimation (ABE) model was introduced in 1997 by Schofield and Shepperd as an alternative for the algorithmic techniques (Shepperd and Schofield, 1997). In this model, the effort value is obtained by comparison of one service with similar and previously completed services (historical cases). In fact, by using Similarity Function, ABE finds the similarities of one service with the similar services (based on the service features) and after selecting some appropriate services (called analogies and shown by KNN parameter) the final solution will be found using Solution Function. The graphic scheme of ABE method is given in Figure 3.

Figure 3 ABE method diagram (see online version for colours)



3.4 *ANN*

The neural network is a nonlinear model that imitates the function of human brain and has frequently been used for estimating effort (Shukla et al., 2014). The neural network consists of a set of neurons in several layers that transport incoming information on their outgoing connections to other units through weighting and by using a suitable transfer function. To generate the output, the inputs take the weight and bias of each neuron and the transfer function processes the inputs of each neuron (Nassif et al., 2012; Pillai and Jeyakumar, 2015).

4 Empirical evaluation

This section explains simulation and comparison of the described methods. The simulation was performed with the help of MATLAB powerful software and the objective was to compare accuracy estimates of machine learning methods.

4.1 Datasets

In order to create and evaluate the associated estimators, two real datasets were used: ISBSG and NASA. The followings are descriptions of each dataset.

4.1.1 ISBSG dataset

ISBSG is a great company located in Australia (ISBSG, 2011). This paper uses the existing data on 11th release of ISBSG dataset which includes partial information of 5,052 software projects. This repository, which uses 109 features for each project, has collected its information from 24 different countries. An appropriate filter is required for selecting an applied and reliable subset of ISBSG projects. In the first step, the project with quality rates other than A and B were removed; therefore there was no doubt in the accuracy of the data. Then, the projects were filtered by some resource level other than development, so that the learning effort and alike are not considered in them (resource level ≤ 1). Finally, the projects that measurement metric of their sizes were other than IFPUG were removed. In the end, by following the above-mentioned filters, 66 software services were obtained and the research was continued on them. Among all the present features, six important ones [Input count, Output count, Enquiry count, File count, Interface count and Adjusted FP] were selected that influenced the development effort [Normalised effort in hours]. Statistical information of ISBSG dataset is given in Table 1.

Table 1 Description of ISBSG dataset

Variable	*Minimum*	*Maximum*	*Mean*	*Median*	*Std.*
InpCont	3	1,185	169	95	199
OutCont	10	698	143	67	165
EnqCont	3	653	150	116	137
FileCont	7	384	129	108	97
IntCont	5	497	76	43	95
AFP	107	2,245	672	507	534
NorEffort	562	60,826	6,860	4,899	8,406

4.1.2 NASA dataset

The second dataset used, are known projects related to NASA that statistical information is presented in Table 2. This dataset was first introduced by Bailey and Basili (1981) and later extensively used in various studies (Elish, 2009; Dejaeger et al., 2012). In this dataset, exist two independent variables development line (DL) and methodology (M) and one dependent variable effort (E). DL variable indicates the number of lines of program development which comments on its considered and M is a combination criteria of methodologies used in software development. E is programming effort and its measurement unit is man-month. This dataset consists of 18 observations which are belonging to 18 NASA software project.

Table 2 Description of NASA dataset

Variable	*Minimum*	*Maximum*	*Mean*	*Median*	*Std.*
DL	2.1	100.8	33.58	17.15	31.67
M	19	35	27.77	18.5	5.23
Effort	5	138.3	49.47	26.2	44.43

4.2 Evaluation criteria

In this study, our objective is compare the accuracy of the method and therefore will use two criteria accepted PRED and MMRE. In addition to the testing results, Is used statistical methods leave one out cross validation (LOOCV) which each project will be used once as a test case and the process is repeated to number of observations. This method is the only method which using it can ensure to validity of the results obtained (Kocaguneli and Menzies, 2013). The use of this technique will increase the validity of the results and the probability that there will be a larger number of random selections. A basic question, and, in fact, the most important parameter in any evaluating and estimation method, is its degree of accuracy: how far the estimated value is from the actual one. Equation (2) shows the relative error (RE) for evaluating the efficiency of a method. In this equation, E is the amount of the actual effort and E' the expected, or estimated, amount (Shepperd and Schofield, 1997).

$$RE = \frac{E' - E}{E} \tag{2}$$

The MRE parameter is an important and commonly used criterion in estimation, and its value for a service is shown in equation (3). In fact, MRE is the absolute error in estimating project, and the lower it is, the more efficient the related method.

$$MRE = \frac{|E' - E|}{E} \tag{3}$$

$PRED(l)$ is another evaluation criterion and shows the percentage of the estimates l% different from the actual value. This parameter is defined in equation (5); in which N is the total number of reviewed studies and A the number of projects with MRE of less than l. The usual value for l is 0.25, in this research too, $PRED(0.25)$ was used. All of the criteria measure the accuracy of the estimation method; however, MMRE must be as small as, and PRED as big as, possible.

$$MMRE = \frac{\sum_{i=1}^{N} MRE}{N} \tag{4}$$

$$PRED(l) = \frac{A}{N} \tag{5}$$

4.3 Feature selection

In this study, we will consider two fundamental issues:

a Do considered total features of the datasets?

b Just choose the features that influence to the effort.

To obtain the relationship between independent variables and dependency the amount of effort, our analysis of the spearman rank order cross correlations did onto variables of each dataset. The results obtained of this analysis for ISBSG and NASA datasets, respectively, in Tables 3 and 4 are shown.

Table 3 Cross correlations analysis for NASA dataset

	DL	*M*	*Effort*
DL	1	0.2715	0.9814
M		1	0.2135
Effort		1	

Table 4 Cross correlations analysis for ISBSG dataset

	InpCont	*OutCont*	*EnqCont*	*FileCont*	*IntCont*	*AFP*	*NorEffo*
InpCont	1	0.6036	0.6581	0.3875	0.2970	0.8779	0.7059
OutCont		1	0.4298	0.4274	0.4219	0.7465	0.4865
EnqCont			1	0.2271	0.1962	0.7360	0.5555
FileCont				1	0.3970	0.6055	0.3074
IntCont					1	0.4710	0.2927
AFP						1	0.6538
NorEffort							1

As you can see, for ISBSG dataset, the main features (the amount of dependency) are respectively InpCont, AFP and EnqCont that in following used for comparison. Also is clear from Table 3 which most effective features of NASA dataset, is DL to value of 0.9814 which shows a strong dependence on the amount of effort. So with the help of this analysis, evaluation and simulation will be done once over the entire dataset and again only on a subset of it.

4.4 Results

Table 5 shows the results of five different methods of machine learning on the ISBSG dataset. In the first part of Table 5 brought effective portion of the dataset and in the second part, the results can be seen on the whole dataset. For each section, both the MMRE and PRED(0.25) criteria are considered.

As you can see, the results of the two parts are different but the difference is not sensible and thus, here, with select features precision of the estimates did not much help to increase. Best results is belongs to SWR method to value MMRE = 0.6759, PRED(0.25) = 0.2424 and worst performance is CART method to value of MMRE = 1.3539, PRED(0.25) = 0.197.

Table 5 Estimation results on ISBSG dataset

	Subset features		*Hole data*	
	MMRE	*PRED(0.25)*	*MMRE*	*PRED(0.25)*
MLR	0.938	0.2576	0.9179	0.2879
SWR	0.6759	0.2424	0.6784	0.2424
CART	1.3539	0.197	1.3909	0.2879
ABE	0.8177	0.303	0.8052	0.3333
ANN	0.9123	0.2727	1.0173	0.303

Finally, Table 6 shows the results of estimation methods on the NASA dataset. Also in this part, Table 6 is composed of two parts the effective data and total data but unlike ISBSG dataset, here the results of these two sections are very different and indeed feature selection is very effective and has high accuracy. The results table reveals that the best performance is belongs to the ANN method with values MMRE = 0.1898, PRED(0.25) = 0.7778.

Table 6 Estimation results on NASA dataset

	Subset features		*Hole data*	
	MMRE	*PRED(0.25)*	*MMRE*	*PRED(0.25)*
MLR	0.2327	0.7222	0.2108	0.8333
SWR	0.2956	0.6111	1.0159	0.3333
CART	0.2764	0.5000	0.2764	0.5000
ABE	0.2984	0.5556	0.9625	0.5000
ANN	0.1898	0.7778	0.2172	0.8333

Figures 4 and 5, respectively, shows chart error rate (MRE values) for each of the services for ISBSG and NASA datasets. Figures 4 and 5 have been drawn on the part of the datasets. The use of LOOCV technique is known are well in this diagram and the performance each method is shown. For NASA dataset have 18 projects and for ISBSG dataset have 66 projects. Skip charts in some places, shows inefficiency and high error estimation method at that point. About ISBSG dataset which number of projects is large, also the error and skip values are more. The number of peaks of each graph have a direct correlation with the PRED(0.25) (25% deviation of the true value). Here, SWR and ANN methods have good uniformity and almost have acceptable error for all services.

The results obtained show that essentially, these methods are different efficiency and this difference in the various types of datasets (historical data) is show. Although three methods of ANN, ABE, and SWR it seems to work better than other estimation methods.

Therefore, researchers should pay attention to the nature of their work to the select the appropriate method.

Figure 4 The MRE values distribution in the various models present in ISBSG (see online version for colours)

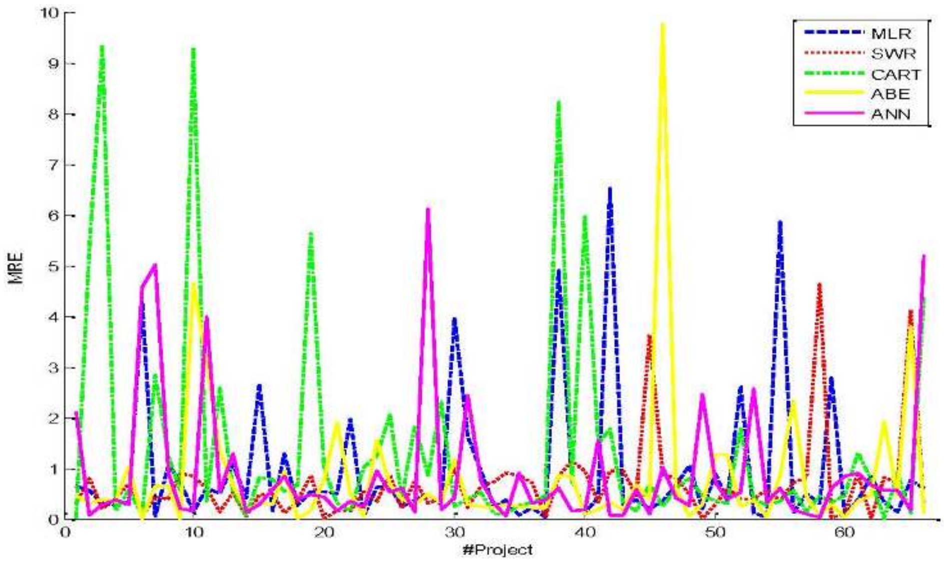
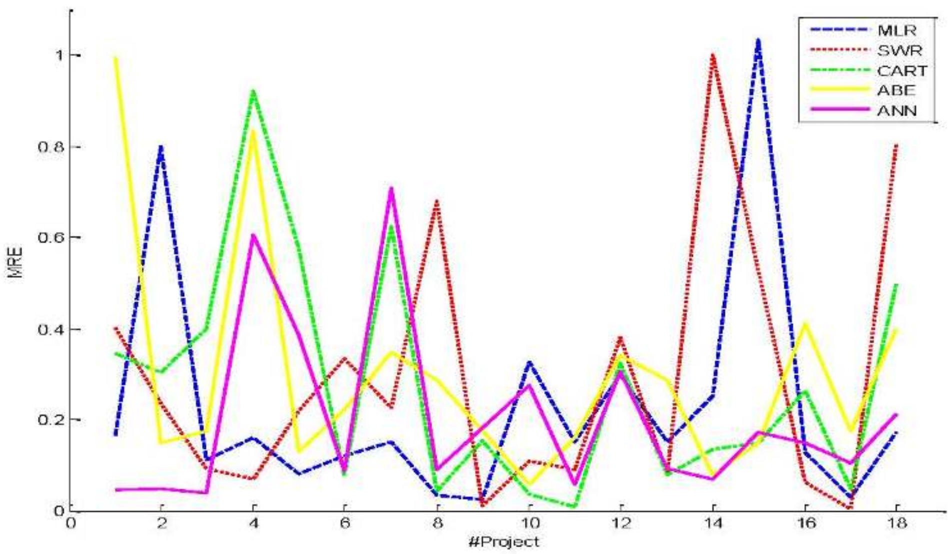


Figure 5 The MRE values distribution in the various models present in NASA (see online version for colours)



5 Conclusions and future work

Accurate estimates of effort essential for software services development, is a major concern both in industry and at academia. So far, several methods have been proposed to estimate the effort that generally placed into three main groups: expert judgement, algorithmic, and machine learning. In this paper, an empirical evaluation of the efficiency and accuracy of machine learning methods was conducted in five important models. Datasets used for this work were NASA and ISBSG and compare criteria were MMRE and PRED(0.25). Also using of dependency analysis method, the main features of each dataset was specified and analysis and comparison on both sections took place fully and effectively. The results obtained, will help to choice and a better understanding of methods for estimating. Future works may be including the new areas of software engineering such as the defect and fault prediction.

References

Ahmed, M.A., Omolade Saliu, M. and AlGhamdi, J. (2005) 'Adaptive fuzzy logic-based framework for software development effort prediction', *Information and Software Technology*, Vol. 47, No. 1, pp.31–48.

Albrecht, A.J. and Gaffney, J.E. (1983) 'Software function, source lines of code, and development effort prediction: a software science validation', *IEEE Transactions on Software Engineering*, Vol. 9, No. 6, pp.639–648.

Araújo, R.D.A., Soares, S. and Oliveira, A.L. (2012) 'Hybrid morphological methodology for software development cost estimation', *Expert Systems with Applications*, Vol. 39, No. 6, pp.6129–6139.

Azzeh, M. (2011) 'Software effort estimation based on optimized model tree', *Proceedings of the 7th International Conference on Predictive Models in Software Engineering*, ACM, Alberta, Canada.

Azzeh, M., Neagu, D. and Cowling, P.I. (2010) 'Fuzzy grey relational analysis for software effort estimation', *Empirical Software Engineering*, Vol. 15, No. 1, pp.60–90.

Bailey, J.W. and Basili, V.R. (1981) 'A meta-model for software development resource expenditures', *Proceedings of the 5th International Conference on Software Engineering*, IEEE Press.

Bardsiri, A.K. and Hashemi, S.M. (2014) 'Software effort estimation: a survey of well-known approaches', *International Journal of Computer Science Engineering*, Vol. 3, No. 1, pp.46–50.

Bardsiri, V.K., Jawawi, D.N.A., Bardsiri, A.K. and Khatibi, E. (2013a) 'LMES: a localized multi-estimator model to estimate software development effort', *Engineering Applications of Artificial Intelligence*, Vol. 26, No. 10, pp.2624–2640.

Bardsiri, V.K., Jawawi, D.N.A., Hashim, S.Z.M. and Khatibi, E. (2013b) 'A PSO-based model to increase the accuracy of software development effort estimation', *Software Quality Journal*, Vol. 21, No. 3, pp.501–526.

Bardsiri, V.K., Jawawi, D.N.A., Hashim, S.Z.M. and Khatibi, E. (2014) 'A flexible method to estimate the software development effort based on the classification of projects and localization of comparisons', *Empirical Software Engineering*, Vol. 19, No. 4, pp.857–884.

Benala, T.R., Mall, R., Srikavya, P. and HariPriya, M.V. (2014) 'Software effort estimation using data mining techniques', *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India*, Vol. 1, Springer.

Boehm, B.W. (1981) *Software Engineering Economics*, Prentice-hall Englewood Cliffs, New Jersey, USA.

Boehm, B.W. and Valerdi, R. (2008) 'Achievements and challenges in Cocomo-based software resource estimation', *Software*, Vol. 25, No. 5, pp.74–83, IEEE.

Boehm, B.W., Madachy, R. and Steece, B. (2000) *Software Cost Estimation with Cocomo II with CDrom*, Prentice Hall PTR, New Jersey, USA.

Breiman, L., Friedman, J., Stone, C.J. and Olshen, R.A. (1984) *Classification and Regression Trees*, Wadsworth and Brooks-Cole Advanced Books and Software, Pacific Grove, California, USA.

Dalkey, N. and Helmer, O. (1963) 'An experimental application of the Delphi method to the use of experts', *Management Science*, Vol. 9, No. 3, pp.458–467.

Dejaeger, K., Verbeke, W., Martens, D. and Baesens, B. (2012) 'Data mining techniques for software effort estimation: a comparative study', *IEEE Transactions on Software Engineering*, Vol. 38, No. 2, pp.375–397.

Elish, M.O. (2009) 'Improved estimation of software project effort using multiple additive regression trees', *Expert Systems with Applications*, Vol. 36, No. 7, pp.10774–10778.

ISBSG (2011) *International Software Benchmarking Standard Group.*

Jeffery, R., Ruhe, M. and Wieczorek, I. (2001) 'Using public domain metrics to estimate software development effort', *Proceedings. Seventh International Software Metrics Symposium, 2001: METRICS 2001*, IEEE.

Jones, C. (2007) *Estimating Software Costs: Bringing Realism to Estimating*, McGraw-Hill Companies, New York.

Jørgensen, M. and Halkjelsvik, T. (2010) 'The effects of request formats on judgment-based effort estimation', *Journal of Systems and Software*, Vol. 83, No. 1, pp.29–36.

Khatibi, V. and Jawawi, D.N. (2011) 'Software cost estimation methods: a review 1'.

Kocaguneli, E. and Menzies, T. (2013) 'Software effort models should be assessed via leave-one-out validation', *Journal of Systems and Software*, Vol. 86, No. 7, pp.1879–1890.

Mendes, E., Watson, I., Triggs, C., Mosley, N. and Counsell, S. (2003) 'A comparative study of cost estimation models for web hypermedia applications', *Empirical Software Engineering*, Vol. 8, No. 2, pp.163–196.

Moløkken-Østvold, K. and Jørgensen, M. (2004) 'Group processes in software effort estimation', *Empirical Software Engineering*, Vol. 9, No. 4, pp.315–334.

Nassif, A.B., Capretz, L.F. and Ho, D. (2012) 'Software effort estimation in the early stages of the software life cycle using a cascade correlation neural network model', *International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012 13th ACIS*, IEEE.

Phannachitta, P., Keung, J. and Matsumoto, K-I. (2013) 'An empirical experiment on analogy-based software cost estimation with CUDA framework', *22nd Software Engineering Conference (ASWEC)*, IEEE, Australia.

Pillai, S. and Jeyakumar, M. (2015) 'General regression neural network for software effort estimation of small programs using a single variable', *Power Electronics and Renewable Energy Systems*, pp.1099–1107, Springer.

Shepperd, M. and Schofield, C. (1997) 'Estimating software project effort using analogies', *IEEE Transactions on Software Engineering*, Vol. 23, No. 11, pp.736–743.

Shukla, R., Shukla, M. and Marwala, T. (2014) 'Neural network and statistical modeling of software development effort', *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012)*, Springer.

Srinivasan, K. and Fisher, D. (1995) 'Machine learning approaches to estimating software development effort', *IEEE Transactions on Software Engineering*, Vol. 21, No. 2, pp.126–137.

Wen, J., Li, S., Lin, Z., Hu, Y. and Huang, C. (2012) 'Systematic literature review of machine learning based software development effort estimation models', *Information and Software Technology*, Vol. 54, No. 1, pp.41–59.

Zhang, W., Yang, Y. and Wang, Q. (2015) 'Using Bayesian regression and EM algorithm with missing handling for software effort prediction', *Information and Software Technology*, Vol. 58, No. 1, pp.58–70.