Empirical Evaluation of Different Machine Learning Methods for Software Services Development Effort Estimation through Correlation Analysis

Amid Khatibi Bardsiri, Seyyed Mohsen Hashemi
IAU
Iran
khatibi_amid@yahoo.com, isru@yahoo.com

ABSTRACT: *The concept of development effort generally means the time or the cost of developing a software service. An essential factor to successfully manage and control a project is the accurate estimation of the development effort and an over and underestimation lead to the loss of project resources. So far, different effort estimation models have been presented in three domains: expert judgment, algorithmic methods, and machine learning methods. Recently, several approaches in the last domain, machine learning, have been applied for software service development effort, which had a higher performance in comparison to the other two domains. This paper presents an empirical evaluation of the performance and accuracy of five main machine learning methods using the correlation analysis approach and investigates the effects of feature selection on the estimation accuracy. The evaluations and comparisons are performed using two well-known and real-world datasets, NASA and ISBSG, and two artificial datasets, Moderate and Severe. Finally, the obtained results provide a clear illustration of the performance of these machine learning methods and the effects of feature selection on the estimation accuracy.*

Keywords: Effort estimation, Software Service, Machine Learning, Empirical Evaluation, Correlation Analysis

Received: 22 September 2015, Revised 27 October 2015, Accepted 2 November 2015

© 2016 DLINE. All Rights Reserved

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 categorized in six groups: Parametric methods such as SEER-SEM, COCOMO (Boehm and Valerdi 2008), Expert judgment such as WBS, Delphi methods (Jørgensen and Halkjelsvik 2010), Learn base models such as ABE (Phannachitta, Keung et al. 2013), Regression methods such as OLS, ROR (Jeffery, Ruhe et al. 2001), Dynamic models, and Hybrid models (Dejaeger, Verbeke et al. 2012).

The introduction of function point (FP) by Albrecht in 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 in 2000(Boehm, Madachy et al. 2000). On the other

hand, because of the inability of algorithmic methods in controlling dynamic behavior of software projects and the lack of complete information of a project in primary stages, non-algorithmic methods have been presented.

Expert judgment 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, Friedman 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, Neagu et al. 2010, Dejaeger, Verbeke et al. 2012, Shukla, 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, Verbeke et al. 2012, Wen, Li 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 organized in 5 sections. The second section reviews the related works. Sections 3 describemachine 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 Judgment: 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, Li 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, Verbeke et al. 2012), COCOMO models and COCOMO II (Boehm 1981, Boehm, Madachy et al. 2000)are the most famous algorithmic models, and ABE (Shepperd and Schofield 1997), CART(Breiman, Friedman et al. 1984), Expert judgment (Dalkey and Helmer 1963)and Artificial Neural Network(Araújo, Soares et al. 2012), Learning and artificial intelligence techniques (Azzeh 2011), fuzzy rules and optimization algorithms (Ahmed, Omolade Saliu 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 5 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, Verbeke et al. 2012, Bardsiri and Hashemi 2014).

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, Jawawi et al. 2013, Bardsiri, Jawawi et al. 2013, Bardsiri, Jawawi et al. 2014):

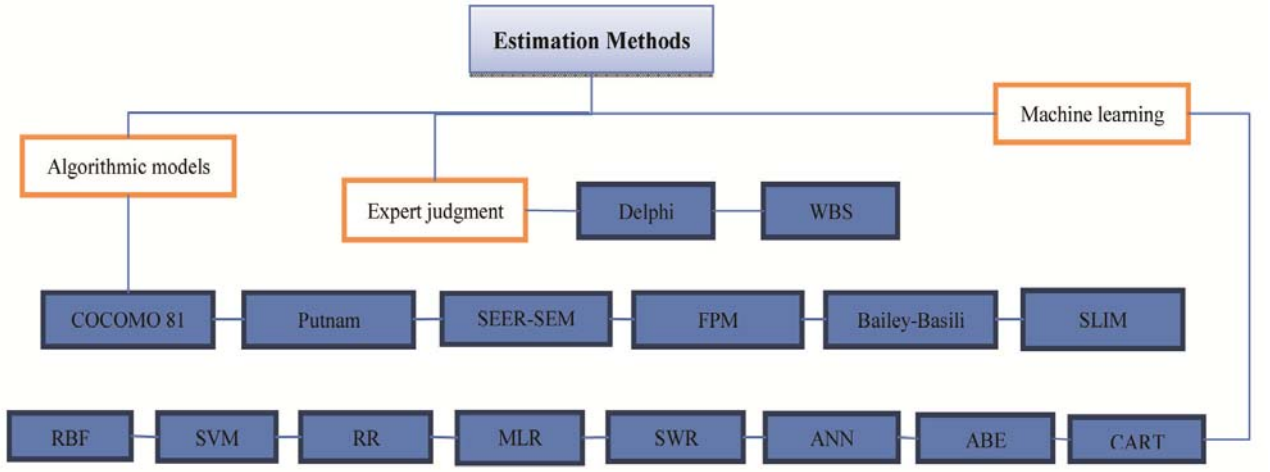

Figure. 1 Various types of effort estimationmethods

$$Y = B_1 X_1 + B_2 X_2 + \ldots B_n X_n + b \qquad (1)$$

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 minimized. MLR (Multiple Linear Regression) and SWR (Step Wise Regression) are examples regression models(Mendes, Watson et al. 2003).

3.2. CART

The purpose in CART(Classification And Regression Trees) 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, Friedman 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, Verbeke et al. 2012, Bardsiri, Jawawi et al. 2013, Bardsiri, Jawawi et al. 2013, Benala, Mall et al. 2014, Zhang, Yang et al. 2015).

3.3 Analogy Based Estimation (ABE)

The 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 bycomparison 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 2.

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, 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, Capretz et al. 2012, Pillai and Jeyakumar 2015). The simple neural network model presented in Figure 3.

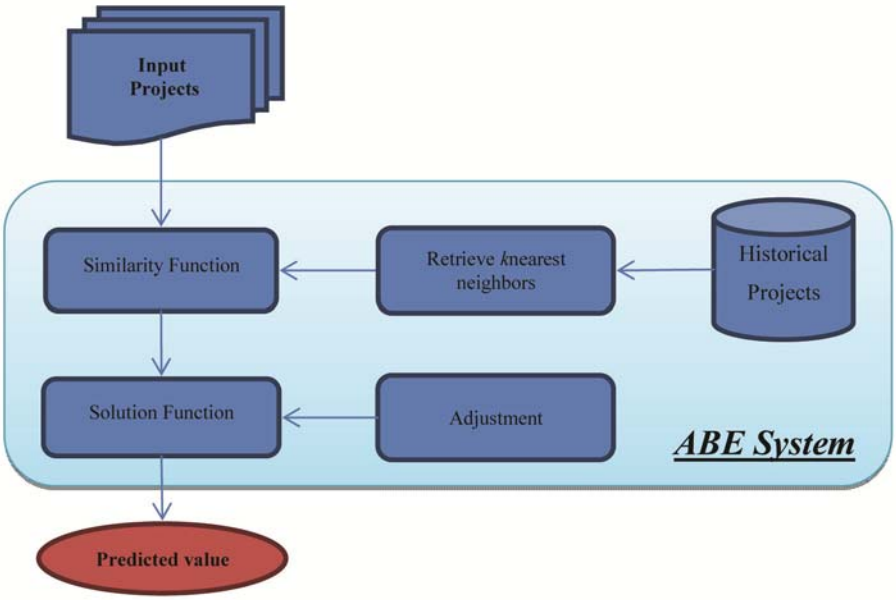

Figure . 2. Analogy Based Estimation diagram

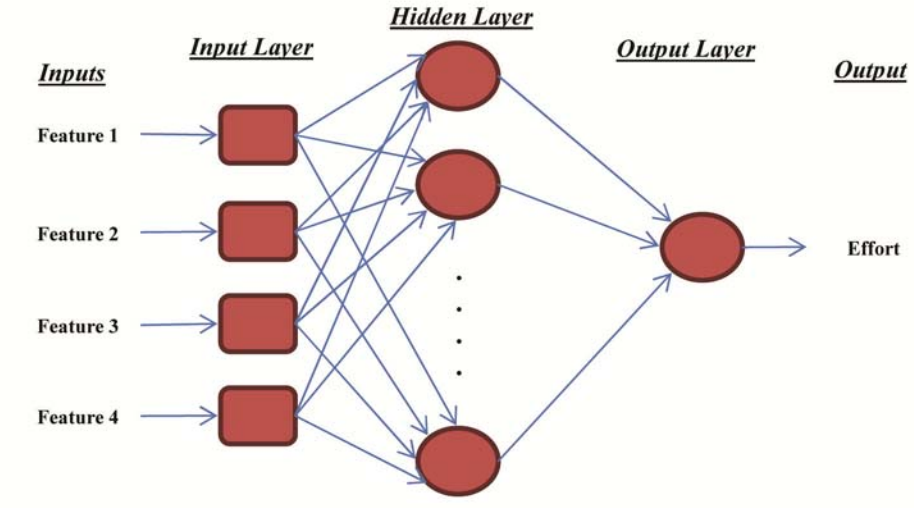

Figure 3. Simple neural network architecture

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, fourdatasets were used, two real datasets: ISBSG and NASA, and two artificial datasets: Moderate and Severe.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 5052 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 function point] were selected that influenced the development effort [Normalized effort in hours]. Statistical information of ISBSG dataset is given in Table 1.

4.1.2. NASA Dataset

The second employed dataset consists of NASA‘s known projects, whose statistical information is presented in Table 2. This dataset was first introduced by Bailey and Basili (Bailey and Basili 1981) and later used extensively in different studies(Elish 2009, Dejaeger, Verbeke et al. 2012). NASA dataset contains 18 observations that belong to its different software projects. This dataset includes two independent variables, *DL* (Development Line) and *M* (Methodology), and a dependent variable, *E* (Effort). Variable *DL* refers to the number of development lines in the application, including comments. Variable *M* is a combinatory measure of software development methodologies and variable *E* indicates software management and programming efforts, measured in man-man units.

Variable	Minimum	Maximum	Mean	Median	Std
InpCont	3	1185	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	2245	672	507	534
NorEffort	562	60826	6860	4899	8406

Table 1. Description of ISBSG 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

Table 2. Descriptionof NASA dataset

4.1.3. Artificial datasets

Unfortunately, real datasets are often old and small with a large number of outliers. Therefore, we are forced to use artificial data to test models. Pickard et al. (Pickard, Kitchenham et al. 2001)introduced a method for generating simulated data. Equation 2 shows the basis of their method (and so do (Ahmed, Omolade Saliu et al. 2005) and (Li, Xie et al. 2009)).

$$Y = 1000 + 6x_1\,sk + 3x_2\,sk + 2x_3\,sk + e_{het} \quad (2)$$

In this equation, $x_1\,sk$, $x_2\,sk$, and $x_3\,sk$ are independent variables obtained from the gamma random distribution of the variables x'_1, x'_2, and x'_3 (with a mean of 4 and variance of 8). The next variable is relative error calculated from the following equation.

$$e_{het} = c \times e \times x_1\,sk \quad (3)$$

In this equation, *e* is the random error resulting from normally distributed random variable with a mean of zero and a variance of 1 and, finally, *c* is a constant. An artificial dataset has the three main features of variance, skewness, and outlier; and two different types of datasets are obtained depending on the values of these three features. Values of the outliers are obtained through multiplication and division of a percentage of the data by specific constant values. Table 3 shows the adjustments made in the two artificial datasets and the values related to them.

Dataset	Relative error constant	Outlier constant	Outliers percentage
Moderate	0.1	2	%5
Severe	6	6	%10

Table 3. Description of artificial data sets parameters

Figures 4 and 5 indicate scatter plots of the 100 data items produced by the datasets Severe and Moderate, respectively. The presence of scattered and deviant values resulting from random distribution causes estimates made by artificial datasets also to be difficult and accompanied by error, so that they can be used in constructing and evaluating models.

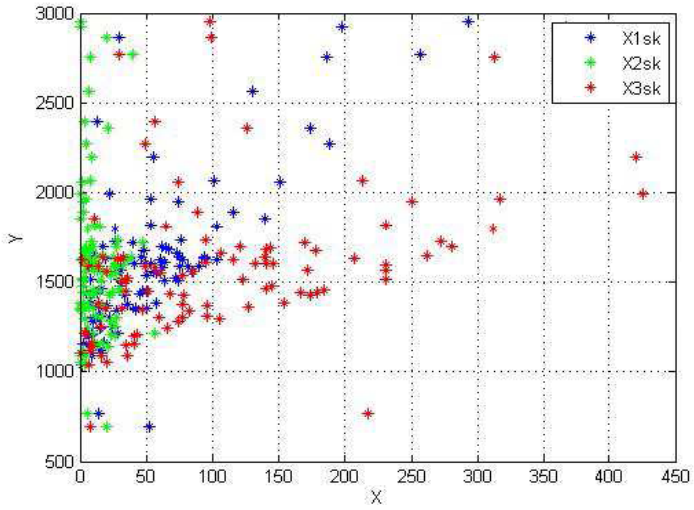

Figure 4. Y versus X values of artificialModerate dataset

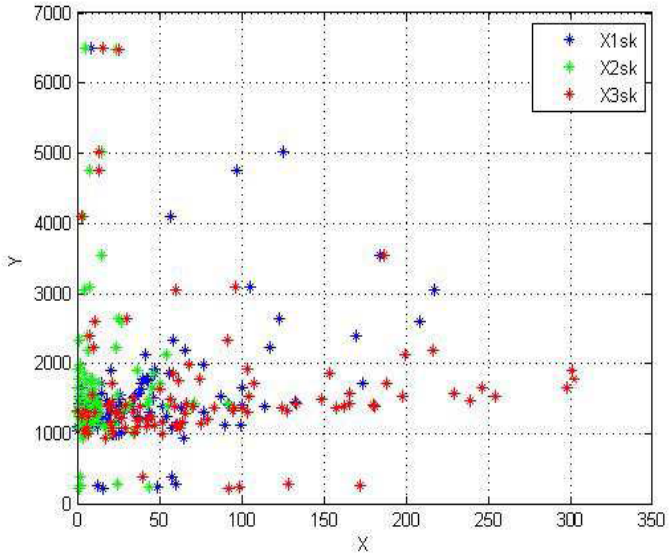

Figure 5. Y versus Xof artificial Severe dataset

4.2. Evaluation Criteria

This study aims to compare the accuracy of different approaches and thus two well-known and accepted measures, PRED and MMRE, are employed. Moreover, the statistical method LOOCV (Leave One Out Cross Validation)is used to validate the results. In this method, each time a project is selected as a test case and the remaining projects are used as training data; this process is repeated for the total number of projects. This approach is the only reliable method of validating the obtained results(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 4 shows the relative error (RE) for evaluating the efficiency of a method. In this equation, *E* is the amount of the actual effort and *E2* the expected, or estimated, amount(Shepperd and Schofield 1997).

$$RE = \frac{E' - E}{E} \tag{4}$$

The *MRE* parameter is an important and commonly used criterion in estimation, and its value for a service is shown in Equation5. In fact, *MRE* is the absolute error in estimating project, and the lower it is, the more efficient the related method.

$$MRE = \frac{|E' - E|}{E} \tag{5}$$

PRED(l) is another evaluation criterion and shows the percentage of the estimates *l* percent different from the actual value. This parameter is defined in Equation7; 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*(0.25) as big as, possible.

$$MRE = \frac{\sum_{i=i}^{N} MRE}{N} \tag{6}$$

$$PRED\ (l) = \frac{A}{N} \tag{7}$$

4.3 Correlation Analysis

Correlation analysis a statistical technique by which the dependency levels between different variables can be achieved. The equations' outputs of this analysis are in range of [-1,1]; a closer value to one indicates a stronger direct relationship between the two variables(Draper and Smith 2014). The advantages of this method include the speed and simplicity of implementation, interpretation, and finding the relationships between the variables and one of its disadvantages is the inability to process categorical variables. Accordingly, in this paper, two basic considerations are made:

a. Do we consider all the features of all datasets?

b. Only features affecting the final effort value are selected. In the latter case, we are applying a type of feature selection that diminishes the size of the dataset.

At 0.05 level of significance, we performed spearman rank order cross correlations analysis on the variables of each dataset to obtain the degree of correlation and dependency between independent variables and the effort(Elish 2009). The high values of this analysis indicate high dependency between variables and the values closer to 0 represent the relative independence of the considered variables. The analysis results using ISBSG, NASA, and Artificial datasets are respectively presented in Tables 4, 5, and 6.

	InpCont	OutCont	EnqCont	FileCont	IntCont	AFP	NorEffort
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**

Table 4. Cross correlations analysis for ISBSG dataset

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

Table 5. Cross correlations analysis for NASA dataset

Each table presents all the potential correlations; however, we are only focused on the effort value. For better comparison, the highest values of each table (most effective features) are marked with star symbol. As we can see, for ISBSG dataset, the most important features (the highest correlation values) are respectively InpCont, AFP, and EnqCont, which are used in the following comparisons. Since this dataset contains a variety of features, we only consider variables with correlation level higher than 0.5. Another interesting point is the considerable correlation of these three variables with one another; according to Table 4, it is clear that there is a significant relationship between the records of this dataset.

Moreover, according to Table 5, the most effective feature of NASA is *DL* with correlation value 0.9814, which indicates strong correlation with the effort value. In addition, the weak correlation between *DL* and *M* (0.2715) indicates their independence. Furthermore, the high value of *DL* shows that simpler estimation models (with less redundant features) can be made for this dataset; in other words, the other feature of this dataset, i.e. *M*, is insignificant in improving the accuracy of the estimation model and thus can be eliminated. Figure 6 presents the distribution and the correlation line of the effort value and the variables of this dataset; it is clear that in contrast to the values of *M*, *DL* values are positioned along a direct line with a positive slope.

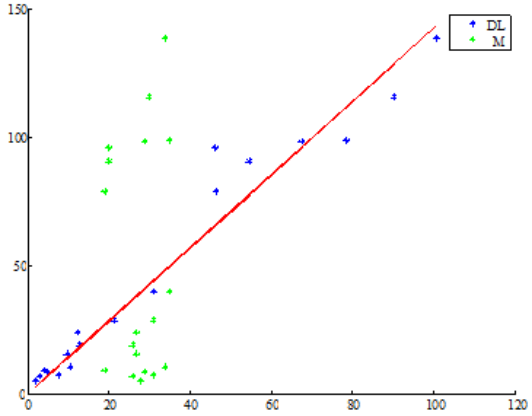

Figure 6. Scatter plot and correlation line for NASA dataset

Finally, Table 6 presents the correlation values of the variables of the artificial datasets. There are negative values in this table

since it is artificial and as it was mentioned in section 4.1.3; their values are created based on gamma random distribution. In spearman rank order cross correlation analysis, the coefficients range is [-1, 1] and negative values indicate the indirect correlation of variables; increasing one proportionally reduces the other variable (consider a direct line with a negative slope)(Green 2014). According to this table, the most important feature of both Moderate and Severe datasets is *X1*, which has the highest correlation with the final variable *Y*. The values of Table 6 are fully in line with Figures 4 and 5 and the scatter and relationships of variables are clearly presented in the illustrated scatter plots.

Therefore, using these analyses, evaluation and simulation was once performed on the whole dataset and again on only its important subset.

	X1		**X2**		**X3**		**Y**	
	Moderate	Severe	Moderate	Severe	Moderate	Severe	Moderate	**Severe**
X1	1	1	-0.1269	0.0959	-0.0421	-0.1141	0.6877*	**0.4048***
X2			1	1	-0.0213	-0.0422	-0.0330	**-0.1758**
X3					1	1	0.4961	**0.1855**
Y							1	**1**

Table 6. Cross correlations analysis for Artificial datasets

4.4 Results

After explaining the fundamentals and definitions in the previous sections, this section evaluates the estimation methods. For each estimation method, MMRE and PRED(0.25) are measured for two cases of the data in its entirety and the effective data. Moreover, for a more precise comparison, for each dataset, the estimation error of each project (MRE) is depicted separately and for all estimation methods performed. As it was mentioned, our validation method is LOOCV that allows the accurate estimation of each project development effort. Table 7 presents the results of five different machine learning methods on the real dataset ISBSG. For each metric (each column), the best solution is marked with star symbol. Moreover, the first part of the table presents the effective subset and the second part shows the results of the entire dataset. For each section, both PRED(0.25) and MMRE are considered and the final section presents the improvement rate of the feature selection approach in comparison to the normal state.

	Subset features		Hole data		Improvement	
	MMRE	PRED(0.25)	MMRE	PRED(0.25)	MMRE	PRED(0.25)
MLR	0.938	0.2576	0.9179	0.2879	-2.18	-10.52
SWR	0.6759*	0.2424	0.6784*	0.2424	0.36	0.00*
CART	1.3539	0.197	1.3909	0.2879	2.66	-31.57
ABE	0.8177	0.303*	0.8052	0.3333*	-1.55	-9.09
ANN	0.9123	0.2727	1.0173	0.3030	10.32*	-10.00

Table 7. Estimation results on ISBSG dataset

The improvement values of different effort estimation methods are obtained by Equation 8 in which variables *In* and *Pr* are respectively the result of the corresponding model on the entire dataset and the output of the same model on the effective subset; the final values are expressed in percent.

$$Improvement_i = \left| \frac{In_i - Pr_i}{In_i} \right| \times 100\% \qquad (8)$$

As we can see, results of the two first sections of the table are different; however, this difference is insignificant and thus, feature selection has not increased the estimation accuracy here. Other than ANN showing an improvement of 10%, feature

selection even may reduce performance. The best result belonged to SWR with MMRE=0.6759 and PRED(0.25)=0.2424 and the worst results were CART with MMRE=1.3539 and PRED(0.25)=0.197. Negative values in the improvement (last) column (i.e. reduction in the estimation accuracy) indicate the inefficiency of the feature selection approach. According to the aggregation and generality of the results of Table 7, a graphical chart is required to separately delineate error values. Figure 7 presents MRE values regarding to the estimation effort of each project. This figure depicts all 66 software services and all five estimation methods on the effective part of the dataset. It is clear that SWR is more uniform and has few fluctuations; while for CART, there are many peaks that indicate the low accuracy of this estimation method.

Furthermore, Table 8 presents results of different effort estimation methods using NASA dataset. This table also contains two sections for the entire and the effective data; however, in contrast to ISBSG dataset, results of these two parts are somewhat different. In fact, feature selection has proved to be effective and enhanced the estimation accuracy. Results of this table indicate that in all cases, the best performance belongs to ANN with MMRE = 0.1898 and PRED(0.25) = 0.7778. Using this dataset, ANN is significantly different and more accurate than other methods. The worst performance belonged to SWR with MMRE = 0.2956 and PRED (0.25) = 0.6111. Regarding this method, the important point is that the high improvement in accuracy using feature selection, higher than 70%, can be explained by its methodology, i.e. using regression, since the number of variables in the regression equation is significantly reduced (to half).

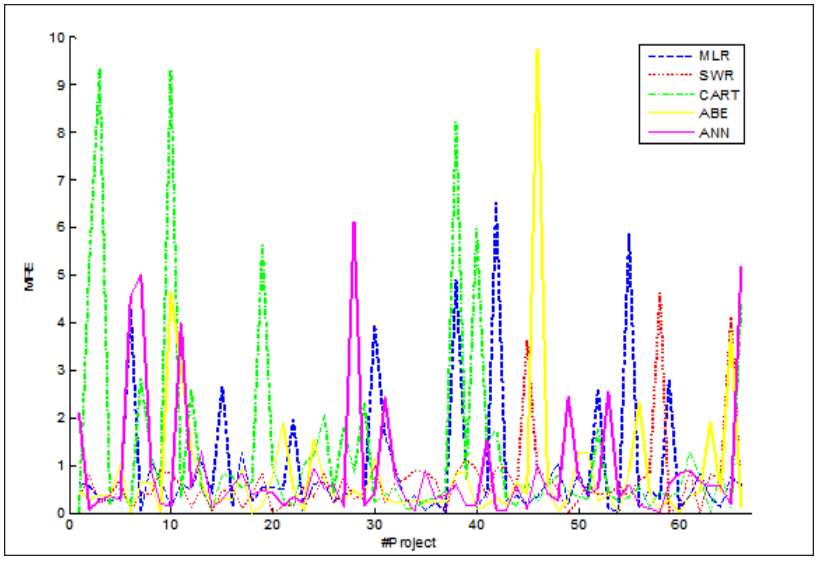

Figure 7. The MRE values distribution in the various models present in ISBSG

	Subset features		Hole data		Improvement	
	MMRE	PRED(0.25)	MMRE	PRED(0.25)	MMRE	PRED(0.25)
MLR	0.2327	0.7222	0.2178	0.8333	-10.38	-13.33
SWR	0.2956	0.6111	1.0159	0.3333	70.90*	83.34*
CART	0.2764	0.5000	0.2764	0.5000	0.00	0.00
ABE	0.2984	0.5556	0.9625	0.5000	68.99	11.12
ANN	0.1898*	0.7778*	0.2172*	0.8333*	12.61	-6.66

Table 8. Estimation results on NASA dataset

In total, according to the values of the improvement column, in contrast to ISBSG dataset, feature selection has a positive effect on the estimation accuracy. Figure 8 presents error values (MRE) for all projects and effort estimation methods. LOOCV statistical approach is clear in this figure and the performance of all five estimation methods is presented on the 18 projects of NASA. In fact, each estimation method is executed 18 times using different values. Due to the small data values

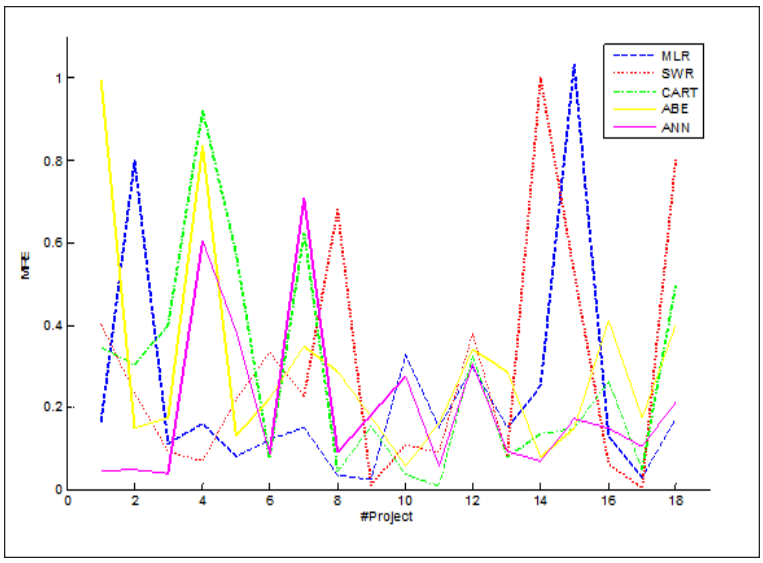

Figure 8. The MRE values distribution in the various models present in NASA

in comparison to the previous dataset, error values and their range is smaller (almost a tenth of ISBSG dataset). The fluctuations in some points of the chart indicate the inefficiency and high estimation errors at those points; the number of peaks in each chart has a direct relationship with PRED (0.25) (deviation from 25% of the actual value). Here, ANN is relatively uniform and shows an acceptable errors value for all projects; while, ABE has many peaks and high heterogeneity in its estimations.

Table 9 follows the same structure of the previous datasets for Moderate artificial dataset. As it was mentioned in section 4.1.3, this dataset was generated using mathematical equations and with low heterogeneity. The best answers in all cases belong to MLR approach, whose results were predictable due to the nature of this estimation method. MLR approximates mathematical equations and since this dataset includes the same equations types, its estimation accuracy is increased. In contrast, SWR with MMRE = 0.8239 and PRED (0.25) = 0 provides an unacceptable answer. Regarding the low data heterogeneity, the answers here are better than real datasets and the error ranges are very small, since in real datasets, skewness, outliers and heterogeneity is much more than artificial datasets. Another important point is that all values of the improvement column were significantly reduced. In fact, feature selection in this dataset has considerably reduced the estimation accuracy of all methods. The reason is that variable *X3* is eliminated, since according to the analysis of section 4.3, this variable has a high dependency with final variable *Y* (Correlation = 0.4961).

	Subset features		Hole data		Improvement	
	MMRE	PRED(0.25)	MMRE	PRED(0.25)	MMRE	PRED(0.25)
MLR	0.1299*	0.94*	0.0424*	0.96*	-206.36	-2.08
SWR	0.8239	0.00	0.7038	0.00	-17.06*	0.00*
CART	0.1592	0.82	0.1007	0.92	-58.09	-10.86
ABE	0.1529	0.85	0.0885	0.94	-72.76	-9.57
ANN	0.1312	0.89	0.0441	0.96	-197.50	-7.29

Table 9. Estimation results on Moderate dataset

Moreover, Figure 9 presents the distribution of error values for all projects of the Moderate artificial dataset. SWR method is completely separate and higher than the other estimation method (higher error values), as the worst approach. Therefore, the high error rate (MRE values) increases the distance of the estimated value from 25% of the actual one, which explains the obtained zero for PRED(0.25) in Table 9. According to the homogeneity of this dataset, almost all methods performed uniformly regarding error values and few peaks and fluctuations are observed in the figure. This is also confirmed by the high values of PRED(0.25).

Finally, Table 10 presents MMRE, PRED(0.25), and the improvement rate of artificial dataset Severe. It is clear from the results that ABE and MLR were accurate and SWR again had the worst answer with MMRE = 0.7872 and PRED (0.25) = 0.2. According to the structure and nature of ABE, the presence of more features and the high inhomogeneity of data have increases its efficiency in comparison to other methods. Moreover, this is also confirmed by the negative values in the improvement column corresponding to ABE. For SWR, feature selection made no difference in the obtained results; in fact, due to using regression equations, this method operates only based on the main variable (The selected variable:*X1*). Another point is the relative improvement of other methods through feature selection. In contrast to dataset Moderate, due to the high inhomogeneity of data, eliminating additional features has helped with the speed and accuracy of the estimation methods. Moreover, regarding the analysis of the third section, other than variable *X1*, which is highly correlated with final variable *Y*, other variables (i.e. *X2* and *X3*) had low correlation with variable *Y* and their removal was not significantly effective in the efficiency of the estimation methods.

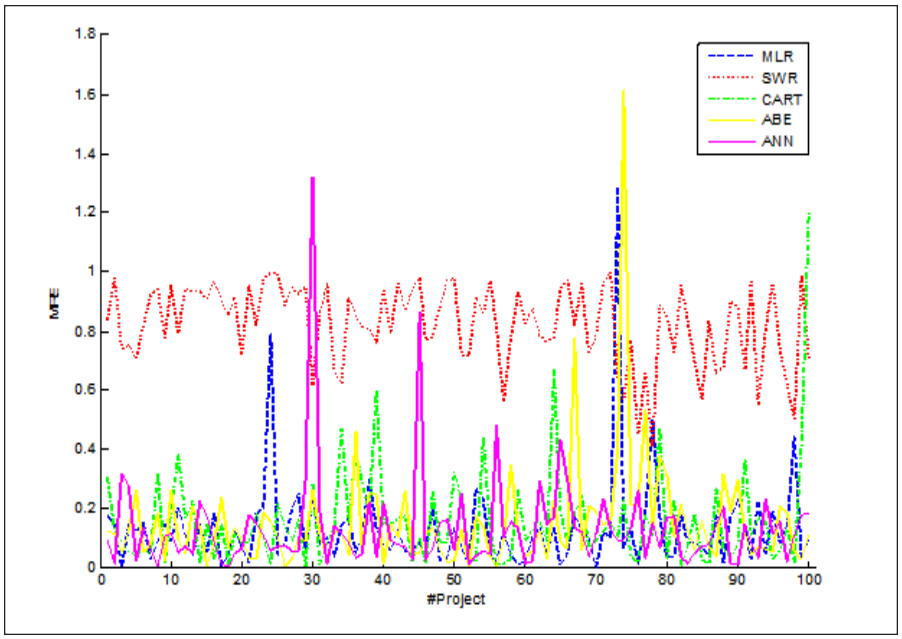

Figure 9. The MREvalues distribution in the various models present in Moderate

	Subset features		Hole data		Improvement	
	MMRE	PRED(0.25)	MMRE	PRED(0.25)	MMRE	PRED(0.25)
MLR	0.4782*	0.62*	0.4971	0.58	3.80	6.89
SWR	0.7872	0.20	0.7872	0.20	0.00	0.00
CART	0.5764	0.54	0.6550	0.53	12.00*	1.88
ABE	0.4786	0.59	0.4137*	0.71*	-15.68	-16.90
ANN	0.4991	0.59	0.5024	0.55	0.65	7.27*

Table 10. Estimation results on Severe dataset

Figure 10 presents the error values for all projects and 5 effort estimation methods. In contrast to the previous dataset, there are many peaks and fluctuations due to the inhomogeneity of the data, which makes the estimation task more difficult. Moreover, due to these peaks, the error range is higher than dataset Moderate (almost 6 times). The interesting point is that there is no peak regarding SWR. The reason is that the equation of this method is close to the equation of generating the dataset. However, the corresponding estimation error is still high.

Results show that the performances of these methods are basically different and this difference is clearly shown for different

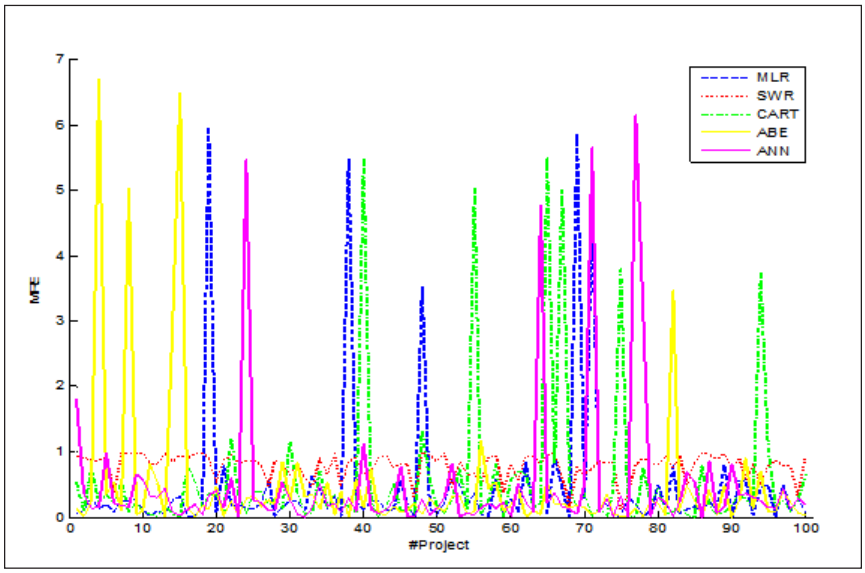

Figure 10. The MREvalues distribution in the various models present in Severe

datasets (historical data). In conclusion, we can say that the three methods ANN, ABE, and MLR seem to outperform other machine learning methods. Moreover, the small size of datasets and selecting important features do not necessarily improve the estimation accuracy of these methods and this is a fact that has been ignored in the previous research. Therefore, researchers should select the appropriate estimation method based on the nature of their work and pay a closer attention to the effective and important features of the dataset.

5. Conclusion and Future Works

The accurate effort estimation for software service development plays a vital role in project management. Over or underestimation both waste system resources and put the company‘s position at risk. In recent decades, several methods were proposed for software service development effort estimation, which are generally categorized in three main group, expert judgment, algorithmic methods, and machine learning methods. According to previous research, machine learning approaches are more accurate and efficient in comparison to the other two groups. Therefore, this paper performed a full empirical evaluation of the accuracy of five well-known machine learning approaches. The datasets used for these evaluations were two real datasets, NASA and ISBSG, and two artificial datasets, Moderate and Severe. Moreover, using correlation analysis, the most important features of each dataset were specified and a comparison was performed on both the entire and effective part of the dataset. The results can help to select a more suitable method for effort estimation and also clarified the effect of feature selection on the estimation accuracy. The future works can include performing this comparison in other similar software engineering domains, e.g. predicting defects and faults.

References

[1] Ahmed, M. A., Omolade Saliu, M., AlGhamdi, J. (2005). Adaptive fuzzy logic-based framework for software development effort prediction, *Information and Software Technology* 47 (1), 31-48.

[2] Albrecht, A. J., Gaffney, J. E. (1983). Software function, source lines of code, and development effort prediction: a software science validation. *IEEE Transactions on Software Engineering*, 9 (6), 639-648.

[3] Araújo, R. d. A., Soares, S., Oliveira, A. L. (2012). Hybrid morphological methodology for software developmentcost estimation, *Expert Systems with Applications*, 39 (6), 6129-6139.

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

[5] Azzeh, M., Neagu, D., Cowling, P. I. (2010). Fuzzy grey relational analysis for software effort estimation. *Empirical*

Software Engineering 15 (1) 60-90.

[6] Bailey, J. W., Basili, V. R. (1981). A meta-model for software development resource expenditures, *In*: Proceedings of the 5th International Conference on Software engineering, IEEE Press.

[7] Bardsiri, A. K., Hashemi, S. M. (2014). Software Effort Estimation: A Survey of Well-known Approaches. *International Journal of Computer Science Engineering,* 3 (1), 46-50.

[8] Bardsiri, V. K., Jawawi, D. N. A., Bardsiri, A. K., Khatibi, E. (2013). LMES: A localized multi-estimator model to estimate software development effort. *Engineering Applications of Artificial Intelligence,* 26 (10), 2624-2640.

[9] Bardsiri, V. K., Jawawi, D. N. A., Hashim, S. Z. M., Khatibi, E. (2013). A PSO-based model to increase the accuracy of software development effort estimation. *Software Quality Journal,* 21(3), 501-526.

[10] Bardsiri, V. K., Jawawi, D. N. A., Hashim, S. Z. M., 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,* 19 (4), 857-884.

[11] Benala, T. R., Mall, R., Srikavya, P., Hari Priya, M. V. (2014). Software Effort Estimation Using Data Mining Techniques. ICT and Critical Infrastructure, *In*: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I, Springer.

[12] Boehm, B. W. (1981). Software engineering economics.

[13] Boehm, B. W., Madachy, R., Steece, B. (2000). Software Cost Estimation with Cocomo II with Cdrom, Prentice Hall PTR.

[14] Boehm, B. W., Valerdi, R. (2008). Achievements and challenges in cocomo-based software resource estimation. Software, IEEE25 (5), 74-83.

[15] Breiman, L., Friedman, J., Stone, C. J., Olshen, R. A. (1984). Classification and regression trees, CRC press.

[16] Dalkey, N., Helmer, O. (1963). An experimental application of the Delphi method to the use of experts. *Management Science,* 9 (3), 458-467.

[17] Dejaeger, K., Verbeke, W., Martens, D., Baesens, B. (2012). Data mining techniques for software effort estimation: A comparative study. *IEEE Transactions on Software Engineering,* 38 (2), 375-397.

[18] Dejaeger, K., Verbeke, W., Martens, D., Baesens, B. (2012). Data Mining Techniques for Software Effort Estimation: A Comparative Study. *IEEE Transactions on Software Engineering,* 38 (2), 375-397.

[19] Draper, N. R., Smith, H. (2014). Applied regression analysis, John Wiley & Sons.

[20] Elish, M. O. (2009). Improved estimation of software project effort using multiple additive regression trees. *Expert Systems with Applications,* 36 (7), 10774-10778.

[21]Green, P. E. (2014). Mathematical tools for applied multivariate analysis, Academic Press.

[22] ISBSG (2011). International Software Benchmarking standardGroup

[23] Jeffery, R., Ruhe, M., Wieczorek, I. (2001). Using public domain metrics to estimate software development effort. *In* Proceedings of the Seventh International Software Metrics Symposium, 2001. METRICS., IEEE.

[24] Jones, C. (2007). Estimating software costs: Bringing realism to estimating, McGraw-Hill Companies New York.

[25] Jørgensen, M. , Halkjelsvik, T. (2010). The effects of request formats on judgment-based effort estimation. *Journal of Systems and Software* 83 (1) 29-36.

[26] Khatibi, V. , Jawawi, D. N. (2012). Software Cost Estimation Methods: A Review 1.

[27] Kocaguneli, E., Menzies, T. (2013). Software effort models should be assessed via leave-one-out validation. *Journal of Systems and Software* 86 (7) 1879-1890.

[28] Li, Y.-F., Xie, M., Goh, T. N. (2009). A study of project selection and feature weighting for analogy based software cost estimation, *Journal of Systems and Software,* 82 (2), 241-252.

[29] Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S. (2003). A comparative study of cost estimationmodels for web hypermedia applications. *Empirical Software Engineering* 8, (2), 163-196.

[30] Moløkken-Østvold, K., Jørgensen, M. (2004). Group Processes in Software Effort Estimation. *Empirical Software Engineering,* 9 (4), 315-334.

[31] Nassif, A. B., Capretz, L. F., 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), 13*th* ACIS IEEE.

[32] Phannachitta, P., Keung, J., Matsumoto, K.-I. (2013). An Empirical Experiment on Analogy-Based Software Cost Estimation with CUDA Framework. 22*nd* Software Engineering Conference (ASWEC), Australian, IEEE.

[33] Pickard, L., Kitchenham, B., Linkman, S. (2001). Using Simulated Data Sets to Compare Data Analysis Techniques Used for Software Cost Modelling. *IEEE Proceedings-Software,* 148 (6), 165-174.

[34] Pillai, S., Jeyakumar, M. (2015). General Regression Neural Network forSoftware Effort Estimation of Small Programs Using a Single Variable. Power Electronics and Renewable Energy Systems, Springer: 1099-1107.

[35] Shepperd, M., Schofield, C. (1997). Estimating software project effort using analogies. *IEEE Transactions on Software Engineering,* 23 (11), 736-743.

[36] Shukla, R., Shukla, M., 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.

[37] Srinivasan, K., Fisher, D. (1995). Machine learning approaches to estimating software development effort. *IEEE Transactions on Software Engineering,* 21 (2), 126-137.

[38] Wen, J., Li, S., Lin, Z., Hu, Y., Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology,* 54 (1), 41-59.

[39] Zhang, W., Yang, Y., Wang, Q. (2015). Using Bayesian Regression and EM Algorithm with Missing Handling for Software Effort Prediction. *Information and Software Technology,* 58. 58-70.